MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY

Predikcie dátových tokov počítačových sietí

DIPLOMOVÁ PRÁCA

Maroš Lipták

Brno, 2013

Prehlásenie

Prehlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Vedúci práce: RNDr. Petr Holub, Ph.D.

Poďakovanie

Rád by som poďakoval RNDr. Petrovi Holubovi, Ph.D. za vedenie tejto práce. Ďalej by som chcel poďakovať Mgr. Tomášovi Rybkovi z firmy SolarWinds za cenné rady pri pravidelných konzultáciách. Moja vďaka tiež patrí Ing. Danielovi Němecovi, Ph.D. a Mgr. Lenke Křivánkovej za odborné konzultácie použitých štatistických modelov.

V neposlednej rade ďakujem za prístup k hardwaru patriacemu do Národnej gridovej infraštruktúry MetaCentrum, poskytnutého v rámci programu „Projekt veľkej infraštruktúry pre výskum, vývoj a inováciu“ (LM2010005).

Zhrnutie

Táto práca sa zaoberá algoritmami vytvárajúcimi bodovú a intervalovú predikciu dátových tokov pomocou štatistických modelov. Pomocou týchto predikcií sú následne detekované anomálie dátových tokov. Algoritmy boli vytvárané v spolupráci s priemyselným partnerom, firmou SolarWinds. Pri navrhovaní algoritmov bol kladený dôraz na jednoduchú konfigurovateľnosť algoritmov, aplikovateľnosť algoritmov na dáta s rôznou charakteristikou a na dostatočnú rýchlosť algoritmov.

Algoritmy boli experimentálne vyhodnotené na dátovej sade siete CESNET2 a tiež na simulovaných dátach. Na základe experimentov sme došli k záveru, že najlepšie výsledky dosahovali algoritmy vyberajúce na základe predchádzajúcich výsledkov najlepší štatistický model, prípadne algoritmy kombinujúce viacero štatistických modelov do váženého priemeru na základe predchádzajúcich výsledkov. Pre vytváranie intervalových predpovedí boli kvôli rozdielnym charakteristikám dátových sád použité empirické predikčné intervaly.

Kľúčové slová

predikcia časových rád, predikčné intervaly, detekcia anomálií, lineárny model, autoregresný model, kĺzavý priemer

Obsah

Zoznam skratiek

AR model	Autoregresný model
AR(p)	Autoregresný model rádu p
MA(p)	Kĺzavý priemer s dĺžkou okna p
ARIMA	Integrovaný autoregresný model s kĺzavým priemerom (Autoregressive integrated moving average)
GARMA	Generalizovaný autoregresný model s kĺzavými priemerm (Generalized Autoregressive Moving-Average)
FARIMA	Frakčne integrovaný autoregresný model s kĺzavým priemerom (Autoregressive fractionally integrated moving average)
SARIMA	Sezónny Integrovaný autoregresný model s kĺzavým priemerom (Seasonal autoregressive integrated moving average)
RMSE	Stredná štvorcová chyba (Root mean square error)
MAPE	Stredná absolútna percentuálna chyba (Mean absolute percentage error)
CUSUM	CUSUM test (cumulative sum control chart))
PCA	Analýza hlavných komponent (Principal Component Analysis)
RSS	Reziduálna suma štvorcov (Residual sum of squares)
AIC	Aikovo informačné kritérium (Akaike information criterion)
BIC	Bayesiánske informačné kritérium (Bayesian information criterion)
PACF	Parciálna autokorelačná funkcia
Cor	Korelácia
SMA	Jednoduchý kĺzavý priemer (Simple moving average)
OLS	Metóda najmenších štvorcov (Ordinary least squares)
SSE	Štvorcová chyba predikcií (Squared errors of predictio)
ROC	Receiver operating characteristic
$\hat{\mathbf{R}}^2$	Adjustovaný index determinácie

Kapitola 1

Úvod

Dátové siete patria v dnešnej dobe k základnej infraštruktúre. Ich správa a monitorovanie sa s rastúcou komplexnosťou stáva čoraz náročnejšie. Vznikajú nové metódy a nástroje, ktoré umožňujú administrátorom rýchlejšie a jednoduchšie spravovať sieť, zisťovať nedostatky, prípadne reagovať na problémy v sieti.

Monitorovanie pozostáva mimo iné aj z analýzy veľkosti dátových tokov na linkách siete. V našej práci sa budeme zaoberať predikciou dátových tokov a následným detekovaním anomálií na dátových linkách. K predpovediam a detekcii budeme používať len historické pozorovania veľkosti dátových tokov. Naše predikcie budú založené na predpoklade, že určité vzory sa v dátovej prevádzke stále opakujú a s určitou pravdepodobnosťou je možné ich predpovedať. Ako príklad takéhoto vzoru si môžeme predstaviť napríklad zvýšenú dátovú prevádzku vo firemných sieťach počas pracovných hodín. K predikciám budeme používať štatistické modely.

Našou prioritou bude vytvorenie algoritmov, ktoré budú použiteľné v monitorovacích nástrojoch počítačových sietí. Budeme sa sústrediť na to, aby naše algoritmy fungovali bez predchádzajúcej konfigurácie od užívateľa. Vzhľadom k rozličnej povahe dátových tokov bude nutné, aby boli algoritmy schopné fungovať na rôznych dátových linkách bez toho, aby užívateľ vyberal, o akú linku sa jedná. V neposlednom rade je dôležité, aby algoritmy neboli príliš výpočtovo náročné, pretože sa predpokladá ich opakovaný beh pri monitorovaní sietí v reálnom čase. Od výsledných algoritmov očakávame, že budú použiteľné v monitorovacom softvéri, ktoré bude spravovať veľké množstvo dátových liniek a zároveň bude mať jednoduchú konfiguráciu. Práca bola vytváraná v spolupráci s firmou SolarWinds, ktorá priebežne poskytovala spätnú väzbu a smerovala prácu k praktickej použiteľnosti.

Predikcie dátových tokov budú slúžiť ako analytický nástroj, ktorý bude umožňovať administrátorovi predchádzať budúcim problémom s dátovou sieťou. Umožnia mu odhaliť možné budúce slabiny siete, prípadne vopred identifikovať slabé miesta v sieti. Detekovanie anomálií bude slúžiť k automatickej detekcii problémov siete spočívajúcich v náhlom náraste alebo poklesu dátových tokov.

Práca pozostáva z šiestich kapitol. V kapitole 2 sa venujeme analýze existujúcich riešení pre predikcie a detekovanie anomálií v dátových sieťach, ich výhodami a nevýhodami a možnosťou aplikácie v našom prípade. Kapitola 3 obsahuje analýzu dátovej sady zo siete CESNET2, ktorú sme neskôr použili pri experimentoch s navrhnutými algoritmami. Návrhu samotných algoritmov vytvárajúcich predikcie a detekujúcich anomálie sme sa venovali v kapitole 4. Pre overenie funkčnosti a porovnanie algoritmov bolo spustených niekoľko experimentov spočívajúcich v simulácii predikcie dátovej prevádzky v sieti CESNET2. Tieto experimenty sa nachádzajú v kapitole 5. V poslednej šiestej kapitole zhŕňame naše výsledky v závere.

Kapitola 2

Súčasné riešenia pre predikcie dátových tokov a detekcie anomálií

2.1 Predikcie dátových tokov

Predikcie na sieťach je možné vykonávať rôznymi technikami. Pri výbere vhodnej techniky sú kľúčové informácie, ktoré máme o sieti dostupné. Ak máme dostupné len dátové toky na sieti, môžeme túto problematiku poňať ako predikciu časových rádov. Autori Lou a Ansari [27] použili pri predikcii dátových tokov autoregresný model, ktorého parametre odhadovali pomocou metódy najmenších štvorcov. Predikované dátové toky boli ďalej použité pre alokáciu optickej dátovej linky. Pre autoregresný model sa autori rozhodli z dôvodu jeho nízkej výpočtovej náročnosti, adaptivite modelu a aplikovateľnosti modelu aj bez predchádzajúcej znalosti štatistického charakteru dát. Autoregresný model sme sa rozhodli implementovať aj v našej práci, zaoberáme sa ním v kapitole 4.1.4.

V práci [16] bol použitý pre predikcie autoregresný model, GARMA model, FARIMA model a naivná predikcia spočívajúca na základe predikovania budúcej hodnoty podľa poslednej nameranej hodnoty. Pre vyhodnotenie experimentov boli použité dáta zo sieťovej prevádzky v Bellcore, algoritmy simulovali predikcie a tie boli následne porovnané s budúcimi hodnotami. Podobným spôsobom budeme robiť experimenty aj v našej práci. Na základe experimentov dosahoval najlepšie výsledky model FARIMA, autoregresný model dosahoval podobné výsledky. Autori ocenili pri autoregresných modeloch rýchlosť a výpočtovú jednoduchosť (v porovnaní s modelom FARIMA). Z týchto výsledkov konštatujeme, že modely triedy ARIMA a FARIMA nie sú vhodné pre naše účely – sú príliš výpočtovo náročné a podobné výsledky je možné dosiahnuť aj s rádovo jednoduchším autoregresným modelom.

Autori Feng a Shu [14] používajú pre predikcie časových radov modely ARIMA, FARIMA a neurónových sietí. Model FARIMA a neurónová sieť dosahovali podobné výsledky, výsledky modelu FARIMA boli nepatrne lepšie. Nevýhodou oproti neurónovým sieťam bola príliš vysoká výpočtová náročnosť, konkrétne bolo potrebných 1500 sekúnd na predikciu 100 hodnôt pomocou FARIMA modelu.

Autori Rutka a Lauks [34], [35] použili pre predikcie dátových tokov neurónové siete, svoje experimenty predikovali na reálnych dátach z návštevnosti webových serverov a simulovaných dátach, pričom časové rady simulovali ako Poissonovský proces. Neurónové siete používané autormi boli pomerne komplikované, v jednom prípade nebolo kvôli náročnosti výpočtov možné dokončiť simuláciu. Neurónové siete sú použité aj v práci autorov Xiang et al. [52]. Mimo použitej neurónovej siete bolo v práci použité ešte zaujímavé kritérium pre hodnotenie predikčných chýb – kumulatívne predikčné chyby. Výhodou oproti kritériám ako RMSE a MAPE je v prípade vykreslenia do grafu možnosť hodnotiť, ako rýchlo sa algoritmus učí (podľa sklonu krivky kumulatívnej predikčnej chyby). Neurónové siete boli ďalej použité aj autormi Madden, Gary, Tan a Joachim [28]. Konkrétne modely vhodné k predikcii autori vytvárali sami. V našej práci budeme implementovať algoritmus, ktorý bude vytvárať

automaticky celý model bez interakcie užívateľa (vzhľadom k požiadavku priemyselného partnera SolarWinds na jednoduchú konfigurovateľnosť). Neurónové siete tiež nebudeme implementovať vzhľadom k ich mierne vyššej výpočtovej náročnosti (ďalšia požiadavka priemyselného partnera – dostatočná rýchlosť algoritmu). Do budúcna by bolo zaujímavé pridať túto predikčnú techniku do nami implementovaného simulátora predikcií. V prílohe H uvádzame postup, ako implementovať novú predikčnú techniku. Neurónové siete budú priamo integrovateľné aj do algoritmu vyberajúceho najlepší model na základe hodnotiacej funkcie popísanej v kapitole 4.1.8.

V prostredí gridov bola pre predikciu dátových tokov použitá metóda podpornej vektorovej regresie (Support vector regression – SVN) v práci [20]. Túto prácu považujeme za zaujímavý prístup k riešeniu problematiky. Za problematickú považujeme optimalizáciu hyperparametrov, tento proces by mohol byť výpočtovo náročný. Do budúcna by bolo zaujímavé porovnať kvalitu predikcií s nami navrhnutými predikčnými metódami.

Zaujímavým prístupom je transformácia časového radu dátového toku, čím môžeme odstrániť niektoré typy nelinearity dát. Tento prístup aplikovali autori Hinich a Molyneux [19]. V našom prípade síce môžeme použiť transformáciu dát a aplikovať ich v lineárnom modeli, nemáme však dostatočnú znalosť o dátach na určenie vhodnej transformácie.

V sade softvérových nástrojov slúžiacich pre predikcie v distribuovanom prostredí NWS [51] bol implementovaný celý rad predikčných techník ako napríklad kĺzavé priemery a exponenciálne priemerovanie [50]. Tiež bol implementovaný algoritmus vyberajúci najlepšiu predikčnú techniku na základe predchádzajúcich výsledkov. Variantu tohto algoritmu budeme implementovať aj v našej práci. Pri výbere predikčnej techniky boli na algoritmy kladené podobné požiadavky ako máme my v tejto práci, pri výbere vhodnej techniky pri implementácii nami navrhnutých algoritmov odporúčame čitateľovi nahliadnuť do výsledkov tejto práce. V porovnaní s touto prácou sme analyzovali aj autoregresné modely a autoregresné modely s umelými premennými.

Ďalším možným prístupom je analýza paketov a vytváranie predikcií na základe znalosti dát, ktoré prechádzajú sieťou. To bolo použité napríklad autormi McGarry et al. pri analýze dátového toku videa. Autori využili znalosť paketov a formát videa, ktoré je prenášané. Svoj postup nazvali ako FFBI (Feed Forward Bandwidth Indication) [29]. Navrhovaný postup je aplikovateľný len na video prenášané po dátovej linke. My sa budeme zaoberať len algoritmami, ktoré nebudú závislé na type paketov prenášaných dátovou linkou.

Pri predikcii je tiež možné využiť znalosť topológie siete. Tento postup využili autori Song et al. [41], modelovali sieť ako strom, na ktorom robili predikcie dátových tokov [42]. Znalosť topológie siete môže prispieť k predikčným schopnostiam modelu. Preto považujeme tento prístup za perspektívne zlepšenie štandardných modelov. Na druhú stranu boli použité modely komplexnejšie a k predikcii je nutné väčšie množstvo dát. V našej práci sme sa modelmi so znalosťou topológie siete nezaoberali. Zo zadania priemyselného partnera, firmy SolarWinds, vyplynulo, že topológia monitorovanej siete nemusí byť vždy známa.

Za zaujímavé považujeme využitie predikcií pre simulovanie zmien pravidiel QoS (Quality of Service) v [5]. Podobne by v budúcnosti mohli byť rozšírené aj lineárne modely použité v tejto práci. Nebolo by možné simulovať všetky pravidlá, model by na druhú stranu bol veľmi jednoduchý a výpočtovo nenáročný.

Väčšina prác sa venuje krátkodobým predpovediam využitia siete (v intervale niekoľkým minút až niekoľkých týždňov). Pre projektovanie siete a jej budúceho využitia môže byť praktická aj dlhodobá predpoveď siete. Pri týchto predpovediach už nebývajú použité štatistické modely, budúce dátové toky bývajú predpovedané na základe expertných predpo-

kladov rastu využitia siete, zvyšovania počtu zariadení a tiež zvyšovania náročnosti služieb v siete. Tento prístup aplikoval napríklad autor Sevcik vo svojej práci [38].

2.2 Detekcie anomálií na sieťach

Podobne ako v prípade predikcie dátových tokov na sieťach je možné poňať problematiku detekcie anomálií v sieťach z viacerých uhľov pohľadu v závislosti na dostupných dátach o sieti. V našom prípade sa budeme zaoberať detekciou anomálií len na základe historických dát dátových tokov na danej linke. Jedná sa o detekciu anomálií na univariatných časových radách. V práci [17] bol na tento problém použitý SARIMA model, jeho parametre boli vyberané minimalizáciou kritéria BIC. Následne bola z bodovej predikcie SARIMA modelu vytvorená minimálna a maximálna hranica hodnôt, ktoré boli tolerované ako normálne hodnoty. Hodnoty mimo tohto intervalu boli označené ako anomálie. Konkrétne bola pre stanovenie intervalov stanovená hodnota 8,5% z NRMSE (Normalized root-mean-square error). Toto kritérium je založené na predikčných chybách. Z tých sú stavané aj empirické predikčné intervaly, ktoré sme použili v tejto práci. Preto očakávame veľmi podobné výsledky metódy založenej na modeli SARIMA a metódami použitými v tejto práci.

Autori Ahmed et al. použili pre detekovanie anomálií kernelovú regresiu a navrhnutý algoritmus KOAD (Kernel-based Online Anomaly Detection) testovali na dátach sieťovej prevádzky a tiež na dátach z cestnej premávky v meste Quebec [2] [1]. Algoritmus KOAD vyzerá veľmi sľubne, pri jeho navrhovaní mali autori podobné priority ako máme my. V budúcnosti by bolo zaujímavé porovnať na reálnych dátach s označenými anomáliami nami uvedené algoritmy s algoritmom KOAD.

V práci [25] bola použitá diskrétna vlnová transformácia, z ktorej bol následne spočítaný interval 3σ pre detekciu anomálií. Obdobu tohto postupu sme implementovali aj my v kombinácii s MA modelom. Výhodou tohto prístupu je veľmi jednoduchá implementácia, výpočtová nenáročnosť a relatívne nízky počet falošne detekovaných anomálií. Nevýhodou je neschopnosť detekovať sezónne vplyvy a závislosť na parametrizácii násobenia metriky σ pre výpočet intervalu na detekovanie anomálie.

Veľmi zaujímavý detektor anomálií bol uvedený v práci [9]. Jedná sa o Holt-Wintersonov model detekcie anomálií založený na exponenciálnom priemerovaní. Výhodou exponenciálneho priemerovania v porovnaní s jednoduchými kĺzavými priemermi spomínanými v našej práci môže byť väčšia váha nových pozorovaní v porovnaní so staršími pozorovaniami. Problémom je opäť správne nastavenie parametrov modelu. V práci sú síce doporučené základné konfigurácie parametrov, tie však nie sú vhodné pre všetky charakteristiky časových radov dátových tokov. Autori v práci ďalej implementujú túto metódu a na základe nej aj detektor anomálií do softvéru Cricket[1] a RRDtool[2], ktorý slúži na analýzu časových radov dátových tokov.

Systém MINDS (Minnesota Intrusion Detection system) využíva k detekovaniu anomálií algoritmus LOF (Local Outlier Factor) [6]. Tento algoritmus počíta pre každé dva body (napr. namerané hodnoty dátového toku) vzdialenosť a hľadá vyčnievajúce body (outliers). Následne je pre každý bod spočítaný index LOF, ktorý udáva, ako moc daný bod vyčnieva. Ak sú hodnoty indexu LOF príliš vysoké, dané pozorovanie je označené ako anomália. Nevýhodou tohto prístupu je výpočtová náročnosť. Algoritmus počíta vzdialenosť medzi

1. http://cricket.sourceforge.net/
2. http://oss.oetiker.ch/rrdtool/

každými dvoma bodmi a jeho zložitosť je $O(n^2)$.

Ďalšie spôsoby detekovania anomálií založené na znalosti dát v sieti spomenieme len okrajovo. Zaujímavý postup detekovania útokov na základe analýzy TCP SYN a TCP FIN paketov je popísaný v práci [48], pre detekciu anomálií je použitý CUSUM test. Popis ďalších techník slúžiacich k detekcii anomálií a útokov na sieťach spolu s popisom techník získavania prevádzkových dát o počítačových sieťach poskytujú autorky Thottan a Ji [44]. Často používaným systémom pre detekciu anomálií je systém SNORT [33] slúžiaci na analýzu sieťovej prevádzky a detekovanie útokov. Detekovaním anomálií pomocou vopred stanovených pevných pravidiel na štruktúru paketov od jednotlivých odosielateľov sa zaoberala práca [13], pričom na detekciu anomálií analyzovali TCP pakety. Analýzou TCP paketov a následnou detekciou anomálií sa tiež zaoberala práca [26]. Analýzu siete na jednotlivých uzloch pomocou PCA použili autori v práci [21], detekciu anomálií robili centrálne z údajov zaslaných jednotlivými uzlami. Analýzou adries uzlov siete, portov ktoré tieto uzly využívajú a koreláciou medzi anomáliami sa zaoberala práca [31]. V práci [32] autori pomocou agentov detekujú anomálie jednotlivých tokov na uzloch siete a následne ich vyhodnocujú vlastným trust modelom. Prehľad ďalších použiteľných techník pre detekciu anomálií v siete poskytuje práca [22].

Kapitola 3

Charakteristika dát

Algoritmy predstavené v tejto práci sa budú zaoberať predikciou časových radov dátových tokov. Algoritmy budú zakladať svoje predpovede na základe predchádzajúcich dátových tokov. Pred začatím navrhovania predikčných algoritmov je nutné aspoň rámcovo poznať charakteristiku časových radov, ktoré budú algoritmy predpovedať. Preto venujeme nasledujúci text analýze časových radov dátových tokov zo siete CESNET2.

Žiaľ navzdory pôvodným očakávaniam sa nám nepodarilo získať dátovu sadu od priemyselného partnera, firmy SolarWinds, s ktorým bola práca vypracovávaná. SolarWinds totiž nemá z právnych dôvodov prístup k dátam o prevádzke na sieťach svojich zákazníkov.

3.1 Popis dátovej sady CESNET

Pre naše simulácie budeme používať časové rady z prevádzky siete CESNET2[1] počas rokov 2007 až 2012. CESNET2 je národná vysokorychlostná počítačová sieť určená pre vedu, výskum, vývoj a vzdelávanie. Prepojuje najväčšie univerzitné mestá Českej republiky a Akadémiu věd České republiky. [40] Dátová sada obsahuje 521 časových radov, ktoré zaznamenávajú vyťaženie jednotlivých sieťových liniek (veľkosť dátových tokov posielaných cez tieto linky). Dáta boli získané zo sieťových prepínačov a smerovačov cez protokol SNMP. Časové rady majú nasledujúcu granularitu záznamov (periodicita jedného záznamu): 1 minúta, 4,6 minúty, 5 minút, 20 minút, 1 hodina, 2 hodiny, 4 hodiny, 8 hodín, 1 deň a 1 týždeň, pričom dĺžka najdlhšej časovej rady je 21 záznamov, najdlhšia časová rada má 1303 záznamov.

Cieľom tejto analýzy je zoznámenie sa s charakteristikou dát a nájdenie určitých spoločných znakov a vzorov v časových radách. Na základe vzorov v dátach následne identifikujeme vhodný model, ktorý využijeme pri vytváraní algoritmu predikujúceho tok v sieti.

Pre hľadanie vzorov a charakteristík môžeme použiť viacero postupov. My sme sa najprv zamerali na analýzu grafov časových radov. Vybrané časové rady uvádzame v prílohe B. Z grafov sme došli k záveru, že v niektorých časových radách je možné identifikovať určitú sezónnosť. Napríklad v dátach s granularitou 1 záznam za dve hodiny (Graf B.1 a B.2) sa nachádza denná sezónnosť, každý deň stúpa počas dňa dátový tok a počas noci klesá. Sezónnosť má v tomto prípade periódu 12 záznamov. V dátach s granularitou 1 záznam (Graf B.3) za jeden deň je možné pozorovať určitú sezónnosť s periódou opakovania 7 záznamov, teda týždennú sezónnosť. Tieto sezónnosti budú pravdepodobne spôsobené zvýšením vyťažením dátovej siete v pracovné dni počas pracovných hodín. Podobná sezónnosť sa ale nenachádza vo všetkých dátach. Napríklad na grafe B.4 nie je možné pozorovať žiadnu podobnú sezónnosť. Jedná sa totiž o linku, na ktorej nie je bežná prevádzková záťaž. Dátová sada obsahuje aj ďalšie dátové linky, v ktorých sa nenachádzala sezónnosť. Typicky sa jednalo o záležné a servisné linky, prípadne dátové linky využívané pre experimentálne a výz-

1. http://www.ces.net/network/

kumné účely.

Na základe týchto pozorovaní môžeme konštatovať, že v niektorých časových radoch sa nachádza určitá sezónnosť, pričom jej periodicita závisí na granularite záznamov časového radu. Prítomnosť sezónnosti prisudzujeme využitiu dátovej linky (pracovné dni...). V niektorých časových radoch nie je možné nájsť žiadnu sezónnosť.

Pri analýze grafov časových radov sme sa zamerali aj na hľadanie trendu v jednotlivých časových radoch. Neboli sme schopní identifikovať žiaden lineárny trend, prípadne trend opísateľný polynómom vyššieho rádu, ktorý by sa nachádzal vo viacerých časových radoch. To zodpovedá približne našim očakávaniam, vzhľadom ku kratšej dĺžke časových radov. Pri dlhších časových radoch zachytávajúcich niekoľko rokov by sme očakávali na niektorých linkách rastúci trend (vzhľadom k neustálemu zvyšovaniu použitia dátových sietí).

3.1.1 Analýza autokorelácie

Pre účely algoritmu predikujúceho dátový tok, ktorý bude zahrňovať aj sezónnosť budeme potrebovať identifikovať sezónnosť v časových radoch na základe určitého kritéria. Ako vhodné kritérium by mohla slúžiť autokorelácia medzi jednotlivými pozorovaniami v časovom rade. Autokorelačná funkcia predstavuje veľkosť korelácie medzi pozorovaním a jeho k-tým opozdením [18]. Rozhodli sme sa analyzovať autokorelačnú funkciu pre jednotlivé časové rady.

Výsledky autokorelačnej funkcie sme si vykreslili pomocou korelogramu, na ktorom sú znázornené výsledky autokorelačnej funkcie pre jednotlivé časové opozdenia. V časových radoch, v ktorých sme na základe grafov dátových tokov ručne identifikovali sezónnosť bola táto sezónnosť identifikovateľná aj vo výsledkoch autokorelačnej funkcie. Ako príklad uvádzame graf s autokorelačnou funkciou C.1 a graf tohto časového radu B.2. Z grafu časového radu sme identifikovali sezónnosť s periódou 12 pozorovaní, čo sa prejavilo aj v grafe autokorelačnej funkcie. Dvanáste opozdenie dosahovalo vysokú koreláciu (to odpovedá vysokej korelácii s 24 hodín pozorovaním). Tak isto dosahovalo vysokú koreláciu aj 24. opozdenie (48 hodín staré pozorovanie).

Naproti tomu v grafe autokorelačnej funkcie C.2 nedosahovalo žiadne opozdenie vyššiu mieru korelácie, čo odpovedá tomu, že sme v grafe B.4 tohto časového radu neboli schopní identifikovať žiadnu sezónnosť.

Analýza ďalších časových radov ukázala, že sezónnosť sa naprieč časovými radmi podstatne líši (aj v časových radoch s rovnakou granularitou záznamov). Nie je možné vyvinúť jeden model, ktorý bude počítať s určitou sezónnosťou a následne ho aplikovať na všetky časové rady. Náš algoritmus bude musieť byť schopný prispôsobiť sa na rôzne periodicity sezónnosti.

3.1.2 Stacionarita časových radov

Ďalším testom, ktorý sme sa rozhodli aplikovať bol test stacionarity. Stacionárny časový rad je taký rad, ktorého štatistické vlastnosti sa počas času nemenia [18]. Niektoré modely pre predikciu časových radov je možné používať len na stacionárnych časových radoch. Iné počítajú aj s určitou nestacionaritou (napr. model ARIMA, ak je časový rad po diferencovaní stacionárna), je však nutné vhodne parametrizovať tieto modely. Preto sme sa rozhodli otestovať, či sú časové rady v našej dátovej sade stacionárne.

Pre test stacionarity sme použili ADF (Augmented Dickey-Fuller test) test [36]. Nulová

hypotéza tohto testu hovorí o tom, či sa v časovom rade nachádza jednotkový koreň. Ak je v časovom rade jednotkový koreň, časový rad nie je stacionárny.

Z celkového počtu 521 časových radov bola nulová hypotéza o prítomnosti jednotkového koreňa zamietnutá z 260 prípadoch na hladine významnosti 0,05. Približne polovica stacionárnych rád je stacionárna, druhá polovica nie je stacionárna. Z tohto testu a z analýzy sezónnosti uvedenej v predchádzajúcej kapitole vidíme, že naša dátová sada obsahuje časové rady s rozdielnymi charakteristikami (ktoré sa líšia aj naprieč granularitami záznamov). Nemôžeme povedať, že by všetky časové rady dátových tokov boli stacionárne, prípadne obsahovali sezónnosť. Preto nie je možné pre predikciu použiť jeden model, ktorý by bol schopný zahrnúť všetky informácie v časovom rade.

3.2 Simulované dáta

Od nášho algoritmu predikujúceho dáta na časových radoch očakávame, že bude schopný predikovať aj na nám doposiaľ neznámych časových radoch. Preto je nutné, aby sme otestovali jeho funkčnosť na čo najširšej dátovej sade obsahujúcej časové rady s rôznymi charakteristikami. Máme k dispozícií pomerne veľkú dátovú sadu zo siete CESNET2, na základe ktorej budeme vyvíjať náš algoritmus. Ak by sme testovali náš algoritmus len na tejto dátovej sade, hrozilo by, že náš algoritmus by sme optimalizovali len pre túto dátovú sadu a na iných dátových sadách by jeho predikcie boli nepresné.

Preto sme sa rozhodli vytvoriť niekoľko modelov, na základe ktorých budeme simulovať časové rady tokov v dátových sadách. To nám neskôr umožní simulovať predikcie na oveľa väčšom množstve časových radov s rôznymi charakteristikami a parametrami (stacionárne, nestacionárne, so sezónnosťou, bez sezónnosti, s vysokým náhodným šumom a podobne).

Taktiež nám to neskôr umožní simulovať anomálie na dátových sieťach a ich detekciu. Časové rady z dátovej sady siete CESNET2 totiž neobsahujú označené anomálie. Bez označených anomálií nie sme schopní následne ohodnotiť algoritmus pre detekciu anomálií (nevieme porovnať, či bola algoritmom detekovaná anomália naozaj anomáliou). Pri simulovaných anomáliách vieme povedať, kedy sa v časovom rade nachádzala anomália a môžeme to následne porovnať s anomáliami detekovanými algoritmom.

Podrobnosti o implementácii modelov generujúceho simulované dáta sa nachádzajú v kapitole H.1 a v zdrojových kódoch simulátora. V tejto časti popíšeme procesy na základe ktorých sú dáta generované.

- Proces *AR(1)* – simuluje bežný dátový tok s pamäťou (nová hodnota dátového toku závisí na predchádzajúcej), k novej hodnote je pridaná vždy náhodná zložka (reprezentujúca inovácie). Proces je parametrizovateľný cez parametre modelu simulátora. Pre tento proces sme sa rozhodli na základe analýzy dát zo siete CESNET, v ktorých sme objavili podobnosť nových pozorovaní dátových tokov s historickými pozorovaniami. Autoregresným modelom a procesom sa venujeme bližšie v kapitole 4.1.4.

- Biely šum – do časového radu je pridávaný náhodný šum generovaný z normálneho rozloženia. Náhodný šum reprezentuje náhodné nepredpovedateľné výkyvy v dátach. Pridávaním náhodných výkyvov testujeme citlivosť algoritmov detekujúcich anomálie voči falošne detekovaným anomáliam.

- Náhodné šoky – do časového radu sú generované náhodné šoky, pričom intenzita šokov je modelovaná pomocou exponenciálnym rozdelením a intenzita šokov je vo-

pred nastavená ako parameter modulu. Tento proces simuluje dáta s charakteristikou ako má napr. dátová linka v grafe B.4, čiže stály dátový tok s nepravidelným náhlym zvýšením dátovej prevádzky na krátke obdobie (ktoré nie je anomáliou, pretože odpovedá využitiu siete a v dátach sa opakuje).

- Model sezónnosti je generovaný pomocou funkcie sínusu (v intervale $(0, \pi)$). Tým simulujeme bežnú dennú sezónnosť. Dopoludnia stúpa využitie dátovej linky, vrchol dosahuje počas obeda a poobede zase klesá.

Pri simuláciách boli využívané kombinácie procesov generujúcich dáta. Napríklad dáta simulované pomocou *AR(1)* modelu s náhodným šumom a sezónnosťou pomocou funkcie sínus simulovali bežný denný vplyv s náhodným šumom a periodicitou. Pri simulácií detekcie anomálií boli do dát pridané ešte náhodné šoky.

Pre simulovanie detekcie anomálií bol model s náhodnými šokmi aplikovaný aj na dátovú sadu zo siete CESNET2. Náhodné šoky v tomto prípade predstavovali anomálie. Výhodou tohto postupu bola simulácia detekcie anomálií na časovom rade pozostávajúcej z reálnych dát, pričom nami vygenerované šoky boli označené ako anomálie. Mohli sme teda porovnať, či algoritmus anomáliu detekoval, alebo nie.

Treba pripomenúť, že naše predstavy o anomáliách nie sú momentálne podložené žiadnou analýzou a nemáme skutočné dáta, ktoré by obsahovali označené anomálie. Naše predstavy o anomáliách v dátových tokoch vychádzajú len z predpokladov o javoch, ktoré považujeme za neobvyklé.

Kapitola 4

Predikovanie dátových tokov

Cieľom našej práce je vytvorenie algoritmu, ktorý umožní predikciu dátového toku na sieti na základe historických dát. K tomuto účelu budeme využívať štatistické modely, ktoré budeme ďalej rozoberať v kapitole 4.1. Samotná bodová predikcia nie je pri predpovedi dátových tokov príliš užitočná, pretože nezohľadňuje neurčitosť predpovede. Preto budeme pri predikciách zostavovať aj predikčné intervaly, v ktorých bude možné budúce hodnoty s určitou pravdepodobnosťou očakávať. Na základe predikčných intervalov následne implementujeme algoritmus, ktorý umožní detekovať anomálie v dátovom toku siete.

Na základe funkčných a nefunkčných požiadavkou priemyselného partnera, firmy SolarWinds, sme došli k záveru, že všetky vytvorené algoritmy musia spĺňať nasledovné:

- Jednoduchá konfigurovateľnosť – algoritmy budú nasadené v softvéri, ktorý neočakáva od užívateľa žiadnu znalosť predikčných algoritmov a štatistických modelov.
- Funkčnosť na dátových linkách s rôznou charakteristikou – softvér vyvíjaný firmou SolarWinds je nasadený na širokom spektre sietí a nie je možné vopred jasne stanoviť charakteristiky dátových liniek. Z tohto požiadavku mimo iné vyplýva, že nami navrhnuté algoritmy budú musieť byť schopné bežať na všetkých možných charakteristikách dátových liniek.
- Výpočtová a pamäťová náročnosť algoritmu – algoritmus bude spúšťaný v rámci softvéru monitorujúceho veľké množstvo dátových liniek v reálnom čase. Z tohto dôvodu je nutné, aby boli predpovede a detekcie anomálií vykonávané dostatočne rýchlo.

4.1 Bodová predikcia

Dátový tok v sieti je možné chápať ako univariantný časový rad, teda časový rad obsahujúci len dátové toky na konkrétnej dátovej linke. Opakom univariatného časového radu je multivariatný časový rad. Bodová predikcia slúži predikcii budúcich hodnôt časových radov. Intervalová predikcia a detekcia anomálií popisovaná ďalej v práci bude vychádzať z modelov bodovej predikcie uvedených v tejto kapitole.

Existuje veľké množstvo modelov pre predikovanie časových radov. My sa budeme zaoberať lineárnymi modelmi vzhľadom k našim výkonnostným požiadavkám na použitie predikčných modelov.

4.1.1 Lineárny regresný model

Lineárny regresný model patrí medzi jednoduchšie modely pre predikciu časových radov. Jeho nespornou výhodou je v porovnaní s nelineárnymi modelmi výpočetná nenáročnosť

a jednoduchosť. Na rozdiel od nelineárnych modelov je možné odhadnúť koeficienty modelu analyticky (pomocou metódy najmenších štvorcov), nie je nutné využívať numerické metódy. Práve kvôli jednoduchosti a rýchlosti sme vybrali tento model ako najvhodnejší pre predikcie. Predpokladáme, že pomocou autoregresného lineárneho modelu budeme vedieť zachytiť vzťahy v časových radoch so sezónnosťou. V kapitole 5 sme na základe simulácie zmerali, že výpočet jednotlivých predpovedí sa pohybuje rádovo v stotinách až desatinách sekundy. Medzi nevýhody patrí prirodzene neschopnosť zachytiť len lineárne vzťahy medzi veličinami.

Vzťah (4.1) popisuje všeobecný lineárny model. Premenná Y predstavuje vektor závislej premennej (dependent variable), X maticu vysvetľujúcich premenných, β vektor koeficientov a ε vektor reziduí [37].

$$Y = \beta X + \varepsilon \tag{4.1}$$

Vektor koeficinetov β býva vo väčšine prípadov neznámy. Pri použití lineárneho modelu je možné ho odhadnúť z dostupných dát pomocou metódy najmenších štvorcov, minimalizáciou kritéria RSS (residual sum of squares) [37]. RSS (uvedené vo vzťahu (4.2)) predstavuje sumu štvorcov reziduí, čiže rozdielov medzi hodnotami závislej premennej $\hat{Y}$ spočítanej pomocou modelu a skutočnými hodnotami Y.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.2}$$

Analytické riešenie koeficientov β metódou OLS (Ordinary least squares) uvádzame v rovnici (4.3). Postupom odvodenia tejto rovnice sa nebudeme zaoberať, je uvedený napríklad v [47].

$$\hat{\beta} = (X'X)^{-1}X'Y \tag{4.3}$$

4.1.2 Ďalšie modely

Často používaným modelom pre predikcie časových radov býva model ARIMA [49]. Jeho výhodou je zahrnutie modelu AR (autoregresný model) a MA (model kĺzavých priemerov) v do jedného modelu a tiež aplikovateľnosť na stacionárne a v niektorých prípadoch aj nestacionárne dáta. Model ARIMA (integrovaný autoregresný model s kĺzavým priemerom) sme pre naše účely experimentálne implementovali, ale narazili sme na problém s automatickým výberom koeficientov tohto modelu. Na identifikáciu sú známe napríklad informačné kritéria ako AIC (Aikovo informačné kritérium) a BIC (Bayesiánske informačné kritérium), problematický je ale odhad koeficientov. Model ARIMA je nelineárny (s výnimkou niektorých modelov bez zložky MA zložky ako napr. ARIMA(0,1,1)) a odhad koeficientov je výpočtovo náročný. Pri automatickom výbere modelu pomocou informačných kritérií by bolo nutné otestovať príliš mnoho variant parametrov a pre každý z nich odhadnúť koeficienty. Takýto algoritmus by bol pre naše účely neúnosne náročný.

Ďalšou používanou technikou pri predikcii časových radov bývajú neurónové siete. Pri nich by bol pre naše použitie opäť problém s automatickým konštruovaním neurónovej siete, pretože nemáme vopred žiadnu znalosť o trendoch, sezónnosti alebo iných vzoroch v sieti.

4.1.3 Kritéria pre porovnávanie predikčných modelov

V práci sa budeme zaoberať automatickým výberom vhodného lineárneho modelu. Pre určenie, ktorý model je najlepší budeme používať ako hodnotiacu funkciu kritéria pre meranie kvality predikcií. Preto predstavíme 3 základné kritéria, ktoré sme v našom simulátore implementovali a používali v našich simuláciách.

RMSE

Metrika RMSE (Root mean square error) alebo tiež RMSD (Root mean square deviation) predstavuje rozdiel medzi modelom predikovanými hodnotami (predikciami) a skutočnými hodnotami. Tieto rozdiely sú v niektorých textoch označované ako reziduá, alebo tiež chyby predikcie. [24] Menšie hodnoty RMSE predstavujú lepší predikčný model (predikcie sú bližšie realite), RMSE rovné nule predstavuje absolútne presne predikcie. Za výhodu metriky RMSE môžeme považovať znevýhodňovanie predikcií, ktoré sú ďalej od skutočnej hodnoty (v porovnaní s kritériom MAPE (spomínaným v ďalšej kapitole), kde takéto predikcie nie sú znevýhodňované). Metrika RMSE patrí k častým metrikám používaným pri hodnotení kvality predikčného modelu.

$$RMSE = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)} \tag{4.4}$$

MAPE

Metrika MAPE (Mean absolute percentage error) vychádza podobne ako metrika RMSE z predikčných chýb (reziduí), jej výsledky sú ale priamo interpretovateľné a tým pádom sú intuitívnejšie [43]. Z tohto dôvodu sme túto metriku zahrnuli do nášho simulátora predikcií. Hodnoty tohto koeficientu predstavujú priemernú percentuálnu chybu predpovede. Ak má predikčný model hodnotu metriky MAPE rovnú 0.05, znamená to, že jeho priemerná chyba je 5%. Predikčná chyba má teda veľkosť 5% predikovaného časového radu. Napríklad dátová linka s priemerným tokom 1000 Mbps a hodnotou koeficientu MAPE pre určitý predikčný model 0.1 má predikčné chyby s priemernou veľkosťou 100 Mbps.

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right| \tag{4.5}$$

Napriek tomu, že koeficienty RMSE aj MAPE vychádzajú pri výpočte z predikčných chýb, ich hodnoty sú rozdielne a nie sú 100% korelované. Pri výpočte RMSE sú totiž predikčné chyby umocnené na druhú, zatiaľ čo pri výpočte metriky MAPE sú predikčné chyby v absolútnej hodnote. V RMSE sú teda chyby viac penalizované.

Theil's U

Ďalším možným kritériom pre meranie kvality predpovedí je koeficient Theil's U, v literatúre tiež označovaný ako Theil's U2 (prvá verzia tohto koeficientu obsahovala nedostatky) [4]. Koeficient pozostáva rovnako ako kritérium RMSE zo štvorca reziduí predikcie, ktoré je následne vydelené štvorcom sumy skutočných hodnôt. Výhodou v porovnaní s koeficientom RMSE je normovanosť, koeficient nadobúda hodnoty od 0 do 1. Z tohto dôvodu sme

ho implementovali v simulátore predikcií a považujeme ho za zaujímavú metriku pre porovnávanie predikčných modelov. Koeficient rovný nule predstavuje ideálnu predpoveď s nulovými chybami, 1 predstavuje predpoveď, pri ktorej prediktor pre každú predikciu predikoval hodnotu 0 (teda predikčná technika je rovnako dobrá ako keby sme pre každé budúce pozorovanie predikovali hodnotu 0). Vyššie hodnoty predstavujú horšie predikcie ako prediktor predikujúci 0 pre každú budúcu hodnotu.

$$Theil's\ U = \frac{\sqrt{\left(\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2\right)}}{\sqrt{\sum_{i=1}^{n} Y_i^2}} \tag{4.6}$$

Porovnanie ďalších kritérií (MdAPE, GMRAE, a MdRAE) pre posudzovanie kvality predikcií aj s experimentom na dátach ponúka práca [3].

RMSE Benchmark

Cieľom našich experimentov bude mimo iné nájdenie modelu, ktorý bude schopný predikovať na rôznych typoch dát kvalitné predikcie. Preto musíme spustiť veľké množstvo simulácií predikcií. Pre hodnotenie simulácií budeme používať kritérium RMSE. To nám umožní vyhodnotiť simuláciu jednej dátovej linky a vybrať, ktorý model bol na danej dátovej linke najlepší. Problematická bude agregácia výsledkov z veľkého množstva simulácií rôznych dátových liniek, aby sme rozhodli, ktorý model fungoval najlepšie na rôznych typoch dát. Ak by sme použili len RMSE, získali by sme výsledky RMSE pre každý časový rad (pre každú dátovú linku), z ktorého by sme boli schopní vybrať najlepší predikčný model pre daný časový rad. My chceme nájsť taký model, ktorý mal najlepšie výsledky na všetkých časových radoch.

Preto sme sa rozhodli zaviesť novú metriku RMSE Benchmark (RMSE Bench). Pre každú simuláciu si zvolíme základný predikčný model (budeme ho označovať ako primitívny model) a všetky ďalšie predikčné modely budeme porovnávať ako percentuálne zlepšenie (prípadne zhoršenie) proti tomuto primitívnemu modelu. Ak bude mať model napríklad hodnoty RMSE Benchmarku po určitej simulácii 0,1, čiže 10%, znamená to, že jeho výsledky boli o 10% lepšie ako výsledky primitívneho modelu.

Ako primitívny model sme väčšinou volili model *AR(1)*, prípadne model *MA(20)*. Predikčný model *AR(1)* vytvára predikcie len na základe jedného predchádzajúceho historického pozorovania. Zjednodušene povedané vytvárame predikcie na zajtra len na základe dnešného stavu. Model *MA(20)* zase odpovedá aritmetickému priemeru 20 posledných predpovedí. Modely označujeme ako primitívne pre ich jednoduchú implementáciu a nízku výpočtovú a pamäťovú náročnosť.

Výsledky metriky RMSE Benchmark pre simuláciu na väčšom počte časových rdov vykresľujeme do histogramu pre každý model. Tieto histogramy následne porovnávame medzi sebou. Ak má určitý model veľký počet výsledkov RMSE Benchmarku rovný 0, znamená to, že bol vo veľa prípadoch rovnako dobrý ako primitívny model. Ak je veľa výsledkov rovných hodnotám viac ako 0, vo veľa prípadoch bol model lepší ako primitívny model a ak sú nižšie ako 0, model bol horší ako primitívny model.

Závery, ktoré vyvodíme z výsledkov metriky RMSE Benchmark neplatia univerzálne na všetky možné charakteristiky dátových liniek. Výsledky sú závislé na časových radoch, na ktorých simulujeme predikcie. Ak budeme simulovať napríklad len na periodických dátach

a určitý model bude dosahovať dobré výsledky, neznamená to, že je to model vhodný na všetky dáta. Nemusí dobre fungovať napríklad na dátach s lineárnym trendom. Taktiež záleží na voľbe primitívneho modelu. V našich experimentoch sa budeme spoliehať na to, že do simulácií zahrnieme veľké množstvo typov časových radov z reálnej aj simulovanej prevádzky a naše závery budú platiť aspoň aproximatívne pre časové rady tokov dátových sietí. Ak by sme chceli naše algoritmy používať na iné typy dát ako dátové toky (napríklad časové rady vývoja akcií, meteorologické teploty a podobne), bolo by pre výber vhodného predikčného modelu nutné spustiť všetky experimenty znovu a danom type dát.

$$RMSE\_BENCH = \frac{RMSE\,(New\_model) - RMSE\,(Simple\_model)}{RMSE\,(Simple\_model)} \tag{4.7}$$

4.1.4 AR modely

AR model (autoregresný model) je lineárny model používaný pri analýze a predikcii časových radov. Reprezentuje model, v ktorom závisí budúce pozorovanie určitým spôsobom na predchádzajúcich pozorovaniach. Predpokladáme, že hodnoty dátových tokov budú závisieť na historických hodnotách (dátová linka je využívaná stále rovnakým spôsobom). V porovnaní s MA modelom nepriemerujeme pri predikcii budúcich hodnôt historické pozorovania, každému historickému pozorovaniu priraďujeme inú váhu (ktorú odhadujeme pomocou metódy najmenších štvorcov). Od tohto modelu budeme očakávať, že zachytí lineárne trendy a niektoré formy sezónnosti.

Závislou premennou v tomto modeli je súčasné pozorovanie (označované ako y_t, kde index t predstavuje poradie v časovom rade, alebo tiež časový index) a vysvetľujúcimi premennými sú minulé pozorovania (označované ako y_{t-i}, kde i je opozdenie minulého pozorovania). Všeobecný zápis AR modelu s úrovňovou konštantou uvádzame v rovnici (4.8). [18]

$$y_t = \alpha + \sum_{i=1}^{p} \beta_i y_{t-i} + \varepsilon_t \tag{4.8}$$

V rovnici (4.9) uvádzame príklad odhadnutého AR modelu s koeficientmi. Predpoveď pomocou tohto modelu by spočívala v dosadení minulých pozorovaní (y_{t-1}, y_{t-2} a y_{t-3}), predpoveď je v tomto prípade $\hat{y}_t$.

$$\hat{y}_t = 0.1 + 0.5y_{t-1} + 0.2y_{t-2} + 0.1y_{t-3} \tag{4.9}$$

Automatická identifikácia AR modelu

Pri použití autoregresného modelu je kľúčová voľba autoregresných členov (premenných modelu). Ak by sme do modelu zahrnuli všetky opozdené pozorovania, model by bol výpočtovo náročnejší. Taktiež by sa stal preidentifikovaný a jeho predikčná sila by sa znížila. Preto je nutné pri použití tohto modelu použiť len obmedzené množstvo opozdených členov. Vzhľadom k tomu, že dátové toky v počítačových sieťach majú rôzne charakteristiky, nie je možné vopred určiť optimálny autoregresný model, ktorý by mal produkovať kvalitné predikcie na všetkých dátových linkách. Pre každú dátovú linku je nutné vytvoriť samostatný model, ktorý bude zodpovedať špecifikám danej linky. Preto je nutné vyberať model

automaticky. Pre automatickú voľbu modelu sme navrhli a implementovali 2 rozdielne algoritmy, ktoré neskôr experimentálne overíme a vyberieme lepší z nich.

$$\alpha(1) = Cor\left(z_t, z_{t+1}\right) \tag{4.10}$$

$$\alpha(k) = Cor\left(z_{t+k} - P_{t,k}(z_{t+k}),\, z_t - P_{t,k}(z_t)\right), pre\, k \geq 2 \tag{4.11}$$

Algoritmus 1 Algoritmus identifikujúci AR model pomocou funkcie PACF

1: *new_model*:=new linear model(*constant*)
2: **repeat**
3: *old_model*:=*new_model*
4: *new_model*:=new linear model(*old_model* + AR(max(PACF(*data*))))
5: **until** F-test(*old_model*, *new_model*).p-value ≥ 0.1
6: **return** *old_model*

Ako prvý sme implementovali algoritmus detekujúci AR členy na základe parciálnej autokorelačnej funkcie (skrátene PACF), jeho pseudokód je uvedený ako algoritmus 1. PACF vyjadruje koreláciu medzi časovým radom a jeho n-tým opozdením po odstránení prvého až n-1 opozdenia. Rovnako ako korelačná funkcia nadobúda hodnoty v intervale [-1,1], pričom -1 znamená nepriamu koreláciu, 0 znamená žiadnu koreláciu a 1 znamená priamu závislosť. Pri analýze časových radov sa parciálna autokorelačná funkcia používa pre určenie závislosti časového radu na predchádzajúcich pozorovaniach a určenie vhodného rádu AR modelu.

V rovnici (4.10) a (4.11) uvádzame vzťah pre výpočet parciálnej autokorelačnej funkcie, kde $\alpha(k)$ predstavuje výslednú parciálnu autokoreláciu a k opozdenie. [7]

Algoritmus na základe parciálnej autokorelačnej funkcie vyberie opozdenie s najvyššou koreláciou a pridá ho do modelu ako autoregresný člen. Následne pomocou F-testu vyhodnotí nulovú hypotézu, že priamka vytvorená pomocou druhého modelu lepšie prekladá dáta ako priamka vytvorená pomocou prvého modelu. Nulová hypotéza je zamietnutá, ak je p-hodnota nižšia ako 0.1. V opačnom prípade algoritmus znova vyhodnotí parciálnu autokorelačnú funkciu a vyberie ďalšie opozdenie s druhou najvyššou koreláciou a pridá ho do modelu. Tento postup sa opakuje až dokým nie je nulová hypotéza F-testu zamietnutá.

Tento algoritmus má jeden parameter, maximálny rad parciálnej autokorelačnej funkcie (koľko opozdení má byť vyhodnocovaných). Tento parameter odporúčame nastaviť na základe granularity dát. Pri granularite jeden záznam za hodinu odporúčame nastaviť parameter aspoň na hodnotu 24 (PACF bude vyhodnocovať maximálne 24 opozdení, teda bude vyhodnocovať koreláciu medzi aktuálnou hodinou a rovnakou hodinou predchádzajúceho dňa). Nastavenie vyššieho parametru si vyžiada pri predpovediach väčšie množstvo historických dát. Platí totiž, že pre AR model rádu 24 treba aspoň 24 historických pozorovaní. Pre dáta s granularitou 1 záznam za deň odporúčame parameter maximálneho opozdenia PACF aspoň 7, aby boli zachytené týždenné trendy.

Algoritmus má zložitosť *O(k)*, kde k predstavuje maximálny rad parciálnej autokorelačnej funkcie. Algoritmus pre identifikáciu modelu nie je nutné spúšťať po každej predikcii. Pre šetrenie systémových prostriedkov je možné identifikovať model napríklad každých 40 predikcií. Ak by sme identifikovali model len na začiatku a následne už vykonávali predpovede len podľa tohto modelu, vznikol by problém pri zmene charakteristiky dát – v tom prípade je nutné spustiť znova algoritmus identifikujúci model. Do budúcna by bolo možné doplniť

algoritmus o štatistický test, ktorý by testoval zmenu charakteristiky dát a na základe toho spúšťal algoritmus identifikujúci model.

Algoritmus 2 Algoritmus identifikujúci AR model sekvenčným pridávaním členov

```
1: old_model:=new linear model(constant)
2: for i=1; i≤max_lag; i++ do
3:    new_model:=new linear model(old_model + AR(i))
4:    if F-test(old_model, new_model).p-value ≥ 0.1 then
5:       old_model:=new_model
6:    end if
7: end for
8: return old_model
```

Druhý implementovaný algoritmus (algoritmus 2) sekvenčne pridáva autoregresné členy a pomocou F-testu testuje, či model s novým autoregresným členom lepšie prekladá historické dáta. Algoritmus teda netestuje, či je nový model lepší pre predikcie. Ak model lepšie prekladá dáta, pridá do výsledného modelu daný autoregresný člen a testuje F-test na ďalšom autoregresnom člene. Zložitosť tohto algoritmu je opäť $O(k)$, kde k predstavuje maximálny rad autoregresného členu, ktorý bude pridávaný.

Problémom tohto algoritmu je fakt, že ak bude do modelu pridaný napr. člen *AR(1)*, nemusí byť do modelu pridaný člen *AR(2)*, aj keby model len s členom *AR(2)* lepšie prekladal dáta. Závisí teda na tom, či sú autoregresné členy pridávané od najvyššieho po najnižší, alebo naopak.

4.1.5 MA modely

Modely kĺzavých priemerov bývajú použité pre vyhladzovanie a predikcie časových radov. Kĺzavý priemer je vhodné použiť ak chceme predikovať dátovú linku s konštantným alebo mierne sa meniacim dátovým tokom obsahujúcim náhodné výkyvy. Tieto výkyvy budú kĺzavým priemerom vyhľadené (eliminované). V prípade aplikácie na dátové toky obsahujúce sezónnosť bude vyhľadená aj táto sezónnosť.

MA(k) model (moving average), alebo tiež kĺzavý priemer vytvára z k okolitých pozorovaní priemer. Pri predikciách pomocou tohto modelu sa neberie do úvahy k okolitých pozorovaní, ale k posledných pozorovaní. Existuje niekoľko metód priemerovania, ako napríklad jednoduché priemerovanie (všetky pozorovania majú rovnakú váhu), vážený priemer, prípadne exponenciálne vyhladzovanie (váha jednotlivých pozorovaní exponenciálne klesá). Rovnica (4.12) znázorňuje všeobecný proces kĺzavého priemeru, kde θ predstavuje váhy pre jednotlivé pozorovania. [8]

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{4.12}$$

Model SMA

Pre naše účely sme sa rozhodli použiť jednoduchý kĺzavý priemer (Simple moving average – SMA), ktorý berie do úvahy k posledných pozorovaní a na základe aritmetického priemeru spočíta budúcu predikciu. Výhoda tohto prístupu je v pamäťovej nenáročnosti. Na rozdiel od AR modelov pri predikcii na základe tohto modelu nie je nutné odhaľovať pomocou

metódy OLS koeficienty modelu. Nevýhodou tohto modelu v porovnaní s AR modelmi je nemožnosť zachytiť sezónne vplyvy. [39]

$$\hat{y}_t = \frac{1}{h} \sum_{t=1}^{h} y_{t-i} \tag{4.13}$$

Model SMA s trendom

Ďalšou nevýhodou SMA modelu je nemožnosť zachytiť jednoduché trendy, ako napríklad trvalý rast a trvalé klesanie. Predikcia bude posunutá o veľkosť rastu tohto trendu. Z tohto dôvodu sme sa rozhodli doplniť SMA model o koeficient b, ktorý odpovedá sklonu lineárneho trendu. Koeficient b odhadujeme pomocou metódy OLS. Ak sa v dátach nenachádza žiaden trend, koeficient b by mal mať hodnotu 1, čo odpovedá modelu SMA.

$$\hat{y}_t = b \frac{1}{h} \sum_{t=1}^{h} y_{t-i} \tag{4.14}$$

Automatická identifikácia MA modelu

Pri predikcii pomocou kĺzavých priemerov je nutné správne zvoliť dĺžku kĺzavého priemeru (parameter k), čiže počet historických pozorovaní, ktoré majú byť brané do úvahy pri počítaní predikcie. Pri vopred známej charakteristike dát nie je problém zvoliť tento parameter na základe znalosti sezónnosti. Napríklad pri dátach s granularitou 1 záznam za deň a týždennej sezónnosti by bolo vhodné zvoliť pre predikciu koeficient k 7 a viac. My ale vopred nevieme, akú sezónnosť budú mať dáta, ktoré chceme pomocou našich modelov predikovať. Preto je nutné odhadnúť koeficient k automaticky.

Algoritmus 3 Algoritmus identifikujúci stupeň MA modelu

```
1: old_degree:=1
2: for i=2; i≤max_degree; i++ do
3:    if RSS(SMA(old_degree)) ≥ RSS(SMA(i)) then
4:       old_degree:=i
5:    end if
6: end for
7: return old_model
```

Nami implementovaný algoritmus inkrementálne zvyšuje dĺžku okna SMA modelu (model jednoduchého kĺzavého priemeru) a pomocou kritéria RSS (reziduálna suma štvorcov), prípadne $\hat{R}^2$ testuje, či je nový model lepší ako predchádzajúci model. Jedná sa v podstate o minimalizáciu reziduí podobne ako v prípade metódy najmenších štvorcov.

4.1.6 Modely s umelými premennými

Umelá premenná v lineárnom modely slúži na vyjadrenie binárnej veličiny, ktorú chceme zahrnúť do modelu. Nadobúda hodnoty 0 (nepravda, false) a 1 (pravda, true) v závislosti na určitom stave. Tento stav môže odpovedať sezónnosti – napríklad premenná nadobúda hodnotu 1 každý pondelok (počas ostatných dní bude nadobúdať hodnotu 0). Môže tiež predstavovať dodatočné informácie, ktoré chceme do modelu pridať. Ak by sme v našom

prípade mali viac informácii o sieti, mohla by binárna premenná odpovedať rôznym stavom, ako napríklad či bola na sieti robená údržba a podobne. V tejto kapitole sa budeme venovať umelým premenným vyjadrujúcim sezónnosť. Budeme sa snažiť automaticky zostaviť autoregresný model s umelými premennými, ktoré budú schopné predikovať sezónnosti.

Pre umelú premennú je rovnako ako pre ostatné premenné v lineárnom modely odhadnutý koeficient β. Tento koeficient predstavuje priamy vplyv danej umelej premennej. Ak má odhadnutý koeficient napríklad hodnotu 400 pre umelú premennú vyjadrujúcu sezónnosť, konkrétne napríklad pondelok v týždňových dátach, môžeme to interpretovať tak, že v pondelok bude dátový tok vyšší o 400 oproti normálu.

Podobne ako v prípade AR modelu je kľúčová voľba vhodných umelých premenných, ktoré zahrnieme do modelu. Ak do modelu vložíme príliš veľa umelých premenných, bude model preidentifikovaný a bude mať nízku predikčnú silu. Tiež budeme potrebovať veľké množstvo historických dát, aby sme mohli aplikovať metódu OLS na odhad koeficientov. Pretože opäť nepoznáme charakteristiku predikovaných dát, je nutné, aby sme implementovali algoritmus, ktorý automaticky zvolí najvhodnejšie umelé premenné a zahrnie ich do modelu.

Pri identifikácii vhodných umelých premenných reprezentujúcich sezónnosť treba najprv identifikovať dĺžku sezónnosti, čiže periodicitu, s akou sa budú umelé premenné opakovať. Pri dátach s granularitou 1 deň a týždňovej sezónnosti by táto dĺžka bola rovná 7 (počet dní v týždni). Do modelu by bolo následne pridaných 7 premenných, kde každá by odpovedala jednému dňu týždňa. Prvá z týchto premenných by odpovedala pondelku a nadobúdala by hodnotu 1 v pondelok. Počas ostatných dní by nadobúdala hodnotu 0. Ďalšia premenná by nadobúdala hodnotu 1 len v utorok.

Ak by sme mali dáta s granularitou 1 hodina, mohli by sme do modelu pridať umelé premenné pre dve dĺžky sezónností. Prvá sezónnosť by mala dĺžku 24, pričom každá premenná by odpovedala jednej hodine počas dňa. Druhá sezónnosť by mala dĺžku 168, jednalo by sa o týždňovú sezónnosť. Prvých 24 premenných by odpovedalo prvému dni v týždni.

Lineárny model s úrovňovou konštantou môže obsahovať až (n-1)! umelých premenných (kde n predstavuje maximálnu dĺžku sezónnosti). Pre zachytenie sezónnosti dĺžky p môže byť použitých 1 až p-1 umelých premenných v závislosti na type sezónnosti. Ak je maximálna dĺžka sezónnosti n a do modelu pridáme všetky možné dĺžky sezónnosti spolu s všetkými možnými umelými premennými, dojdeme práve k výsledky (n-1)! umelých premenných. Toto je extrémny prípad modelu, ktorý predstavuje skôr nežiaduci stav. Napríklad na dátach s granularitou 1 deň a sezónnostiach s dĺžkou 1, 2, 3, 4, 5, 6 a 7 mohlo byť do modelu pridaných až 5040 umelých premenných. Pre odhad koeficientov by bolo následne potrebných viac ako 5040 historických premenných, čo pri dátach s granularitou 1 deň predstavuje viac ako 13 rokov. Preto je nutné eliminovať pri identifikácii umelých premenných nesprávne identifikované umelé premenné.

Po identifikácii periodicity sezónnosti v dátach, ďalej v texte označovanú ako k, môžeme pridať do modelu všetkých k premenných. Nie vždy je to nutné. Ak máme napríklad týždňovú sezónnosť a sezónny vplyv sa prejavuje len v druhý deň v týždni, stačí pridať len túto premennú. Získame tak jednoduchší model, ktorý bude výpočtovo menej náročný a pre odhad koeficientov bude vyžadovať menší počet historických pozorovaní.

Do lineárneho modelu s umelými premennými je možné pridať autoregresné členy (uvádza rovnica (4.15)), prípadne je možné pridať umelé premenné aj do MA modelu. V našej experimentálnej časti sme implementovali algoritmus, ktorý najprv identifikoval autoregresný model pomocou funkcie PACF a následne pre tento model identifikoval umelé pre-

menné.

$$D_i = \begin{cases} 0 & \text{pozitívny stav} \\ 1 & \text{negatívny stav} \end{cases}$$

$$\hat{y}_t = \sum_{t=1}^{h} b_i y_{t-i} + b_{i+h} D_i \tag{4.15}$$

Algoritmus postupne pridávajúci umelé premenné

Algoritmus 4 Algoritmus identifikujúci umelé premenné v modeli, umelé premenné sú postupne pridávané a testované F-testom

```
1: old_model:=new linear model(constant)
2: for p=1; p≤max_period; p++ do
3:    for i=1; i≤p; i++ do
4:       new_model:=new linear model(old_model + dummy_variable(p,i))
5:       if F-test(old_model, new_model).p-value ≥ 0.1 then
6:          old_model:=new_model
7:       end if
8:    end for
9: end for
10: return old_model
```

Prvý implementovaný algoritmus (označený ako algoritmus 4) skúša postupne pridávať umelé premenné pre periodicity sezónnosti 2 až n (n je maximálna dĺžka periodicity). Pre danú periodicitu postupne pridáva po jednom umelé premenné do modelu a následne pomocou F-testu testuje, či nový model vysvetľuje historické dáta lepšie ako model bez umelej premennej. Podobne ako pri algoritmoch identifikujúcich AR modely je aj v tomto prípade problém s vysokou náročnosťou algoritmu. Algoritmus postupne otestuje n! modelov (k predstavuje maximálnu dĺžku sezónnosti). Algoritmus je síce výpočtovo náročný, ale nie je nutné ho spúšťať pri každom vytváraní predikcie.

Pre experimentálne účely sme implementovali verziu tohto algoritmu, ktorá porovnáva modely priamo na základe kritéria RSS a vyberie model s nižšou hodnotou RSS.

Algoritmus hromadne pridávajúci umelé premenné

Druhý implementovaný algoritmus (označený ako algoritmus 5) tiež pridáva postupne umelé premenné pre periodicity sezónnosti n až 2 (maximálna dĺžka periodicity), pri jednej iterácii pridá všetky premenné pre danú periodicitu. Následne vykoná pre každú z týchto premenných t-test a vylúči premenné, ktoré nevysvetľujú dostatočne dáta (neprispievajú k tomu, aby model lepšie vysvetľoval historické dáta). Výhodou oproti predchádzajúcemu algoritmu je vyššia rýchlosť (algoritmus musí otestovať len *n-1* modelov). Taktiež je čiastočne eliminovaný smer pridávania premenných (od najvyššej po najnižšiu alebo naopak).

Algoritmus 5 Algoritmus identifikujúci umelé premenné lineárneho modelu, umelé premenné sú pridávané hromadne a testované t-testom

```
1: model:=new linear model(constant)
2: for p=1; p≤max_period; p++ do
3:    for i=1; i≤p; i++ do
4:       model:=model.add(dummy_variable(p,i))
5:    end for
6:    new_model := model
7:    for i=1; i≤p; i++ do
8:       if t-test(model, dummy_variable(p,i)).p-value ≥ 0.1 then
9:          new_model:=model.remove_variable(dummy_variable (p,i))
10:      end if
11:   end for
12:   new_model := model
13: end for
14: return model
```

4.1.7 Algoritmy eliminujúce premenné lineárneho modelu

Doposiaľ sme popisovali algoritmy, ktoré postupne identifikovali AR členy, MA model a umelé premenné. Taktiež sme sa zmienili o možnosti kombinovať tieto modely navzájom. Môžeme napríklad spustiť algoritmus identifikujúci AR členy a následne do takto identifikovaného modelu pridať umelé premenné. Hrozí ale, že model bude obsahovať veľký počet premenných. Tiež hrozí, že niektoré členy modelu (napr. AR členy) už nebudú potrebné pre vytvorenie kvalitných predpovedí, pretože niektoré javy totiž môžu byť lepšie vysvetľované umelými premennými ako AR členmi. Pri identifikácii lineárneho modelu týmto spôsobom totiž záleží na poradí, v akom identifikujem členy lineárneho modelu.

Z tohto dôvodu sme navrhli algoritmy, ktorý postupne eliminuje nepotrebné členy lineárneho modelu, ktoré boli identifikované rôznymi identifikačnými algoritmami. Cieľom týchto algoritmov je zjednodušenie modelu a odstránenie takých premenných modelu, ktoré neprispievajú k lepším predikciám.

Algoritmus 6 Algoritmus pre elimináciu premenných lineárneho modelu pomocou t-testu

```
1: fitted_model:=fit(model)
2: for variable in fit(previous_model).variables do
3:    if t-test(variable).p-value ≤ 0.1 then
4:       model.remove(variable)
5:    end if
6: end for
7: return model
```

Prvý algoritmus odhadne model na historických dátach a následne v jednej iterácii odstráni všetky premenné, ktorých p-hodnota t-testu je menšia ako 0,1. Výsledkom tohto algoritmu je model, z ktorého boli odstránené premenné s nízkou významnosťou.

Tento algoritmus odstraňuje všetky nevýznamné premenné počas jednej iterácie. Vzhľadom k problému kolinearity môže pri takomto postupe dôjsť k odstráneniu významných premenných, ktoré majú vysokú vypovedaciu hodnotu. Preto sme implementovali druhú

variantu algoritmu eliminujúceho premenné lineárneho modelu, ktorá odstraňuje najmenej významné premenné postupne. Táto varianta v cykle identifikuje najmenej významnú premennú a odstráni ju. Následne pomocou kritéria $\hat{R}^2$ ohodnotí, či sa výpovedná hodnota modelu zlepšia oproti pôvodnému modelu s premennou, alebo došlo k zhoršeniu. Ak došlo k zlepšeniu, algoritmus pokračuje ďalej v eliminácii premenných modelu. Ak došlo k zhoršeniu, algoritmus vráti posledný model, ktorý viedol k zlepšeniu.

Algoritmus 7 Algoritmus iteratívne eliminujúci premenné lineárneho modelu pomocou $\hat{R}^2$

```
previous_model:=model
new_model:=model
while R̂²(previous_model) ≤ R̂²(new_model) do
    min_t_test_pvalue:= ∞
    min_variable := null
    for variable in fit(previous_model).variables do
        if t-test(variable).pvalue ≤ min_t_test_p then
            min_t_test_p:= t-test(variable).p-value
            min_variable := variable
        end if
    end for
    previous_model := new_model
    new_model:=previous_model.remove(min_variable)
end while
return previous_model
```

4.1.8 Algoritmy pre výber najlepšieho modelu

Predstavili sme niekoľko modelov, ktoré boli vhodné na určité typy dát. Pri experimentálnom vyhodnocovaní sa ukázalo, že žiaden z modelov nebol optimálny pre všetky typy dátových tokov. Preto sme sa rozhodli implementovať algoritmus, ktorý by bol schopný automaticky vybrať z niekoľkých štatistických modelov ten najlepší a následne pomocou neho robiť predikcie.

Výber z jedného modelu na základe hodnotiacej funkcie

Pre beh tohto algoritmu je nutné vykonať niekoľko predpovedí pomocou všetkých dostupných modelov. Následne sú tieto predikcie porovnané so skutočnými hodnotami a algoritmus spočíta hodnotiacu funkciu (napr. RMSE) pre každý model. Pre ďalšie predikcie vyberie model s najlepšími výsledkami hodnotiacej funkcie.

Algoritmus abstrahuje od použitých modelov, je možné ich ľubovoľne pridávať alebo odoberať. Pre rýchle predpovede je možné použiť len jednoduché MA modely, pre podrobnejšie predpovede je možné zahrnúť aj AR modely a modely s umelými premennými. Taktiež je možné zahrnúť ďalšie modely, ako napríklad model ARIMA alebo neurónové siete (neboli v práci implementované).

Pre beh tohto algoritmu je nutné, aby bol splnený predpoklad, že dáta nemenia svoju charakteristiku. Napríklad ak sa v dátach po určitú dobu vyskytuje sezónnosť, algoritmus bude predikovať pomocou modelu AR s umelými premennými. Ak sa náhle zmení charak-

teristika dát na dáta s konštantným tokom a bielym šumom ($y_t = \mu + \varepsilon$), ktoré sú dobre predikovateľné MA modelom, algoritmus bude ešte niekoľko období predikovať pomocou AR modelu s umelými premennými.

Tento algoritmus musí pred prvou predpoveďou porovnať aspoň 1 predpoveď zo všetkých modelov, z ktorých vyberá so skutočnou hodnotou (potrebuje vypočítať chybu predikcie). To môže byť problém pri granularite dát napríklad 1 deň, pretože oproti ostatným modelom budeme pri tomto algoritme musieť čakať jeden deň naviac na prvú predpoveď. Tento nedostatok je možné kompenzovať tak, že prvá predpoveď bude vykonaná na základe MA modelu a každá ďalšia bude predpovedaná pomocou tohto algoritmu.

Ďalšou nevýhodou je výpočtová náročnosť tohto algoritmu. Pre jeho beh je nutné vykonávať neustále predikciu pomocou všetkých dostupných modelov. Tento nedostatok je možné regulovať pomocou počtu a náročnosti modelov, ktoré zahrnieme do množiny modelov, z ktorých algoritmus vyberá.

Problematická môže byť taktiež pamäťová náročnosť tohto algoritmu. Potrebuje vyhodnocovať predikčné chyby pre každý model. Je teda nutné uchovávať n (počet predikčných chýb, ktoré budú brané v úvahu pri výpočte hodnotiacej funkcie) predikčných chýb v pamäti.

Voľba hodnotiacej funkcie, na základe ktorej bude algoritmus vykonávať voľbu najlepšieho modelu závisí od priorít užívateľa. My sme algoritmus implementovali s dvomi hodnotiacimi funkciami – RMSE a MAPE (rozoberané v kapitole 4.1.3). Funkcia RMSE viac penalizuje väčšie predikčné chyby ako funkcia MAPE. Ak teda užívateľ preferuje modely, ktoré majú menej extrémnych chýb, mal by zvoliť ako hodnotiacu funkciu RMSE. Ak preferuje väčšiu priemernú presnosť algoritmu aj za cenu niekoľko málo extrémne veľkých predikčných chýb, môže zvoliť funkciu MAPE.

V predchádzajúcom texte sme sa zaoberali algoritmami, ktoré zvolili vhodné AR členy do AR modelu, parameter p MA modelu, prípadne vhodné umelé premenné do AR modelu. Všetky tieto problémy by bolo možné vyriešiť pomocou tohto algoritmu. Je možné, že tento algoritmus by dosahoval lepšie výsledky ako nami predstavené algoritmy (nehrozí preidentifikovanosť, pretože hodnotiaca funkcia porovnáva skutočné predikčné chyby). Problém je v náročnosti takéhoto postupu. Algoritmus by musel vykonávať predikcie pre každý jeden okamžik pomocou všetkých možných modelov so všetkými možnými parametrami. Pre MA modely by to nepredstavovalo príliš veľkú výpočtovú záťaž – $O(p)$, kde p predstavuje maximálnu dĺžku kĺzavého okna MA modelu). Tento algoritmus sme dokonca implementovali v kapitole 5.3. Pre model s umelými premennými ale rastie exponenciálne množstvo modelov so zvyšujúcim sa parametrom maximálnej dĺžky sezónnej premennej. Zložitosť tohto algoritmu by rástla exponenciálne. To by predstavovalo problém z pohľadu výpočtovej náročnosti a pamäťovej náročnosti (je nutné ukladať predikčné chyby pre všetky modely). Preto sme pre AR model a AR model s umelými premennými tento variant algoritmu ani nesimulovali.

Vážený priemer všetkých dostupných modelov

Na rozdiel od predchádzajúceho algoritmu tento algoritmus nepredikuje len na základe jediného modelu s najlepšími výsledkami predikčnej funkcie, berie do úvahy viacero modelov na základe výsledkov ich hodnotiacej funkcie. Predikcie robí ako vážený priemer predikcií vybraných modelov (vzorec 4.16), pričom váhy jednotlivým modelom pridelí na základe normovaného výsledku hodnotiacej funkcie (vzorec 4.17). V našich experimentoch

sme používali ako hodnotiacu funkciu v tomto modeli RMSE.

$$y_t = \frac{\sum_{i=1}^{n} w_i y_{mi}}{\sum_{i=1}^{n} w_i} \tag{4.16}$$

$$w_i = \frac{min(\{RMSE(M_j)\}_{j=1}^{N})}{RMSE(M_i)} \tag{4.17}$$

Rovnako ako predchádzajúci algoritmus má aj tento algoritmus nevýhodu vo svojej výpočtovej a pamäťovej náročnosti – neustále musia byť počítané predpovede pomocou všetkých dostupných modelov. Taktiež je nutné na začiatku predikcie vybrať alternatívny model, pretože pre prvú predikciu je nutné poznať predikčné chyby (pre spočítanie hodnotiacej funkcie, aby vedel algoritmus prideliť váhy jednotlivým predikciám). Problematická je aj zmena charakteristiky dát.

V našich experimentoch sme implementovali aj variant tohto algoritmu, ktorý výpočítaval vážený priemer len z obmedzeného počtu modelov s najlepšími výsledkami (v našom prípade sme robili vážený priemer z 3 modelov s najlepšími výsledkami funkcie RMSE). Predchádzajúci algoritmus vyberajúci predikčný model na základe hodnotiacej funkcie predstavuje špeciálny prípad tohto algoritmu vytvárajúceho predikcie na základe váženého priemeru, pričom množstvo modelov braných do úvahy pri výpočte váženého priemeru je obmedzených na 1.

Pre voľbu hodnotiacej funkcie, na základe ktorej sa následne počítajú váhy platí to isté, čo pre predchádzajúci algoritmus, teda je možné vyberať z RMSE, MAPE a iných na základe preferencií užívateľa.

4.2 Intervalová predikcia

Bodová predpoveď síce umožňuje určiť trend, akým sa bude časový rad vyvíjať, pre naše účely však nemá príliš veľký prínos. Predpoveď je vždy spojená s určitou neurčitosťou, ktorá sa môže na dátových linkách rôzne líšiť. Na základe samotnej bodovej predpovede nevieme túto neurčitosť určiť. S bodovou predpoveďou vieme povedať, ako sa časová rad bude vyvíjať, nevieme ale povedať, s akou určitosťou to nastane.

Pre naše účely by bolo oveľa praktickejšie, ak by sme vedeli predikovať rozmedzie hodnôt, v ktorých sa bude dátový tok pohybovať. Na tieto účely slúži predikčný interval. Predikčný interval pozostáva zo spodnej a vrchnej hranice, do ktorej hodnota spadne s určitou pravdepodobnosťou. Predikčný interval (prediction interval) býva označovaný tiež ako intervalová predpoveď (interval forecast), prípadne predikčné hranice (prediction bounds). [11] Predikčný interval s pravdepodobnosťou 95%, alebo tiež 95% predikčný interval predstavuje taký interval, do ktorého s 95% pravdepodobnosťou budú spadať budúce hodnoty.

V prípade vopred známych stochastických procesov a správne vybratého predikčného modelu je možné spočítať predikčné intervaly. My ale nepoznáme proces, na základe ktorého vzniká dátový tok. Z toho tiež vyplýva, že nevieme povedať, či je nami vybratý predikčný model správne zvolený. Postupy pre počítanie predikčného intervalu zvyčajne počítajú s určitým rozložením predikčných chýb (napríklad normálne rozloženie v prípade lineárnych modelov). My sme sa rozhodli analyzovať naše predikčné chyby, aby sme zistili, či nepatria do určitého rozloženia, na základe ktorého by sme mohli predikčné chyby aproximovať.

V dodatku D uvádzame vybrané grafy predikčných chýb pre modely *AR(1)* a *MA(30)* zo simulácií predikcií na reálnych časových radoch prevádzky dátovej siete CESNET2 na

vybraných linkách. Z uvedených histogramov môžeme jednoznačne konštatovať, že predikčné chyby naprieč rôznymi dátovými linkami majú aj pri použití rovnakých modelov rôzne rozloženie. Podobne vyzerali histogramy predikčných chýb z ďalších simulácií predikcií, ktoré v práci z dôvodu priestoru neuvádzame.

Vzhľadom k tomu, že predikčné chyby majú pre rôzne časové rady rozdielne rozloženia, nemôžeme použiť aproximáciu podľa známych rozložení. Ak by sme ju použili, hrozilo by, že predikčné intervaly by boli buď príliš široké, alebo príliš úzke. Pri 95% predikčnom intervale by 95% predpovedí nespadalo do tohto intervalu (ale napríklad len 60% v prípade príliš úzkeho predikčného intervalu, alebo 100% v prípade príliš širokého predikčého intervalu).

V našom prípade predpokladáme, že nami vytvorený algoritmus nebude nasadený na vytvorenie jednej jedinej predpovede, ale bude neustále predpovedať nové hodnoty a budú mu dodávané nové dáta. Predpokladáme, že po určitej dobe behu algoritmu bude mať algoritmus dostatok historických dát. Na základe tohto predpokladu sme sa rozhodli použiť empirické predikčné intervaly.

Empirické predikčné intervaly sú počítané z existujúcich predikčných chýb ako požadované kvantily z týchto predikčných chýb [12]. Základným predpokladom, aby predikčné intervaly fungovali, je proces generujúci dáta s vlastnosťami, ktoré sa nemenia v čase. Výhodou empirických predikčných intervalov je nezávislosť na použitom modeli pre výpočet bodových predikcií a nezávislosť na procese generujúcom dáta. Ďalšou výhodou je automatické prispôsobenie predikčných intervalov na počet období, na ktoré sa počíta predpoveď dopredu, pretože intervaly sa počítajú z chýb predikcií predpovedí na dané obdobie dopredu. Nevýhodou je nutnosť uchovávať predikčné chyby v pamäti a taktiež zo začiatku predikcií nutnosť počkať po určitú dobu (aspoň 50 období), dokým algoritmus nezíska dostatočný počet predpovedí, aby mohol začať vytvárať empirické predikčné intervaly. Empirické predikčné intervaly sa samy prispôsobia aj predikčným chybám, ktoré majú asymetrické rozloženie. To je výhodné najmä v kontexte dátových liniek počítačových sietí, ktoré majú jasné spodné a horné obmedzenie (spodné je 0 a horné je maximálna kapacita linky).

Už sme spomenuli, že pre počítanie empirických predikčných intervalov je nutné uchovávať históriu predikčných chýb. Za minimum považujeme na základe našich skúsenosti uchovávanie aspoň 50 predikčných chýb. Menší počet uchovaných predikčných chýb môže viesť k nepresne spočítaným kvantilom (je nutné ich aproximovať), prípadne k príliš úzkym predikčným intervalom (nebudú zahrnuté dostatočne veľké predikčné chyby). Ak je nutné vytvárať predikčné intervaly aj pri nízkom počte predikčných chýb, odporúčame vytvárať predikčné intervaly s kvantilovými hodnotami zaokruhlenými na desiatky (90% kvantil). V takom prípade nedochádza k aproximácii pri výpočte kvantilu predikčného intervalu. Pokiaľ sa nemenia vlastnosti procesu generujúceho dáta, vo všeobecnosti je možné povedať, že čím viac predikčných chýb si budeme uchovávať, tým lepšie predikcie predikčných intervalov budeme dosahovať. To by ale mohlo byť pamäťovo náročné. Zo skúseností pri simulácii počítania predikčných intervalov sme došli k záveru, že viac ako 300-400 predikčných chýb nemá zmysel uchovávať.

Ďalšou výhodou empirických predikčných intervalov je ich nezávislosť na zvolenom modely pre vytváranie predikcií. Ak zvolíme vhodný model, prípadne model presne odpovedajúci skutočnému dáta generujúcemu procesu a odhadneme správne parametre modelu, po určitom čase (keď bude k dispozícii dostatočný počet predikčných chýb) skonvergujú empirické predikčné intervaly k skutočným predikčným intervalom. Ak nemáme žiadnu znalosť o dáta generujúcom procese a zvolíme nevhodný model, empirické predikčné intervaly budú síce príliš široké (model napríklad nevystihne sezónnosť dát), bude však stále

platiť pravdepodobnosť, s akou budú budúce hodnoty spadať do predikčných intervalov.

V kombinácii s niektorou jednoducho predikčnou technikou ako napríklad kĺzavý priemer s vhodnou šírkou okna môžu byť empirické predikčné intervaly použité ako veľmi jednoduchá a rýchla predikčná technika. Aj v prípade, že dáta generujúci proces neodpovedá modelu kĺzavých priemerov, s empirickými predikčnými intervalmi dostaneme interval, do ktorého budú dáta spadať s danou pravdepodobnosťou. Predikčné intervaly budú síce široké a s lepším modelom by sme dosiahli lepších výsledkov, táto technika bude veľmi jednoduchá na výpočty a bude tiež mimoriadne rýchla (v porovnaní so všetkými ostatnými technikami v práci popisovanými).

Empirické predikčné intervaly je možné skombinovať so všetkými modelmi uvedenými v predchádzajúcom texte. Jediný problém môže nastať pri nestabilných automaticky identifikovaných AR modeloch, teda v prípade, že procedúra identifikujúca model vracia pri každej predikcii rozdielne modely. V takom prípade nie je ale možné automaticky identifikovaný AR model použiť. Túto situáciu je však možné algoritmicky detekovať, dochádza totiž k výraznej zmene modelu AR pri každej identifikácii modelu.

Taktiež je možné skombinovať predikčné intervaly s algoritmom vyberajúcim najlepší model na základe hodnotiacej funkcie. Pri tomto použití je nutné, aby algoritmus počítal predikčné intervaly pre všetky dostupné modely. Algoritmus bude v takom prípade vracať predikčné intervaly modelu, ktorý práve vybral ako najlepší na základe hodnotiacej funkcie bodových predpovedí.

4.3 Detekcia anomálií

Jedným z cieľov tejto práce je vyvinutie algoritmu, ktorý bude vedieť detekovať anomálie v dátových tokoch. Algoritmus by mal byť schopný detekovať anomáliu v momente, keď nastala – jedná sa o takzvanú online detekciu (offline detekciou býva označované detekovanie anomálií v historických dátach).

Za anomáliu budeme v našom prípade považovať náhle zvýšenie dátového toku, ktoré sa vymyká limitom na danom časovom rade. V prípade dátových liniek sa môže jednať o náhle výrazné zvýšenie prevádzky na časovom rade, alebo tiež náhle zníženie prevádzky, napríklad keď linka vypadne. Prínos týchto algoritmov by mal byť najmä pre administrátorov dátových liniek. Na rozdiel od pevne nastavených limitov pre hlásenie anomálií by mali nami navrhnuté algoritmy umožňovať citlivejšie detekovanie anomálií, čo by malo prispieť k skoršiemu identifikovaniu problémov na sieťach.

Rovnako ako v prípade bodových predikcií a predikčných intervalov budeme mať k dispozícii len časové rady tokov na sieťach. Pre detekovanie anomálií budeme používať len historické a súčasné pozorovania dátových tokov, nebudeme používať žiadne ďalšie dodatočné informácie.

4.3.1 Detekcia anomálií pomocou smeradatnej odchýlky

Prvým implementovaným algoritmom je jednoduchý algoritmus detekujúci chyby na základe smerodajnej chyby. Algoritmus spočíta smerodajné odchýlky pre predchádzajúce pozorovania a ak rozdiel medzi novou hodnotou a priemernou hodnotou prekračuje násobok smerodajnej odchýlky, algoritmus vyhodnotí novú hodnotu ako anomáliu. Násobok smerodajnej odchýlky je primárne nastavený na hodnotu 3, čo odpovedá pri normálnom rozložení predikčných chýb pravdepodobnosti 99,865%, že nová hodnota bude nižšia ako trojnásobok

smerodajnej odchýlky. Algoritmus posudzuje pre vyhodnotenie anomálie veľkosť odchýlky od priemernej hodnoty.

Hodnoty časového radu, ktoré boli detekované ako anomálie nie sú neskôr zahrnuté do výpočtu smerodajnej odchýlky. Ak by boli anomálie zahrnované do výpočtov smerodajnej odchýlky, algoritmus by sa postupom času stával menej citlivým na anomálie, pretože smerodajná odchýlka by sa stále zväčšovala. Problémom tohto prístupu nastane v prípade, ak algoritmus detekuje ako anomáliu hodnotu, ktorá nie je anomáliou a predstavuje normálny stav. V takomto prípade bude algoritmus pri ďalších výpočtoch príliš citlivý na anomálie (pretože smerodajná odchýlka je príliš malá).

Tento algoritmus neberie do úvahy časové rady s trendom a sezónnosťou. Dôležitým parametrom tohto algoritmu je nastavenie dĺžky pozorovaní, z ktorých sa počíta priemer a smerodajná odchýlka. Ak bude počet historických pozorovaní príliš nízky (nižší ako dĺžka sezónneho správania), algoritmus môže vyhodnocovať sezónnosť ako anomáliu.

4.3.2 Dvojkroková detekcia využívajúca predikčné intervaly

Druhý navrhnutý a implementovaný algoritmus detekuje anomálie na základe predikčných intervalov. Ak hodnota prekročí 100% empirický predikčný interval pre dané obdobie, algoritmus prejde do stavu 1, možná anomália. Ak sa aj v ďalšom období nachádza hodnota mimo predikčného intervalu, algoritmus prejde do stavu 2, detekovaná anomália. Algoritmus detekuje anomáliu až dokým sa nová hodnota časového radu nenachádza v predikčnom intervale.

Pre počítanie predikčných intervalov je možné použiť rôzne postupy. Ak poznáme proces generujúci dáta, predpovede predikčných intervalov robíme na základe skutočného modelu dát, môžeme predikčné intervaly počítať na základe rozdelenia pravdepodobnosti dát (v tomto prípade treba zvoliť iný ako 100% kvantil intervalu). V takom prípade je možné detekovať anomálie v dátach už od prvej dostupnej predpovede. Ak nepoznáme proces generujúci dáta a predpovede počítame pomocou modelu, ktorý nemusí odpovedať dátam (napríklad AR, MA, automaticky identifikovaný AR alebo algoritmus vyberajúci zo všetkých dostupných modelov), môžeme použiť empirické predikčné intervaly. V tomto prípade je možné detekovať anomálie až po určitom množstve vykonaných bodových predpovedí (30 a viac). Pre výpočet predikčného intervalu je nutné mať dostatok predikčných chýb.

V porovnaní s algoritmom detekujúcim predpovede na základe smerodajnej odchýlky sa tento algoritmus dokáže prispôsobiť trendom a sezónnostiam (ak sa im dokáže prispôsobiť predikčný model). Náročnosť tohto algoritmu je vyššia ako pri predchádzajúcom algoritme. Pre beh algoritmu je nutné najprv spočítať bodovú predpoveď, potom predikčné intervaly a následne rozhodnúť, či sa jedná o anomáliu alebo nie. Najnáročnejšie na tomto procese je počítanie bodových predpovedí.

Ak algoritmus detekuje anomáliu, táto hodnota nebude zahrnutá do hodnôt, z ktorých sa počítajú predpovede a predikčné intervaly. Podobne ako pri predchádzajúcom algoritme by v totiž došlo k zníženiu citlivosti predpovedí. Stavy č. 1, možná anomália sú zahrnuté do dát, z ktorých sa počítajú predpovede, ak za týmito stavmi nenasleduje anomália, teda ak sa možná anomália nepotvrdia. Ak za stavom č. 1 nasleduje anomália, možné anomálie nie sú zahrnuté do dát, z ktorých sa počítajú predpovede. Pri detekovaní anomálie sú pre posúdenie ďalších hodnôt počítané predikčné intervaly pre na n období dopredu (n odpovedá časovej dĺžke trvania súčasnej anomálie). Ak by sme tento postup nerobili, algoritmus by sa pri dlhšie trvajúcich anomáliách postupne stával citlivejší. Pri počítaní predikčných

intervalov bez zahrnutia aktuálnych hodnôt (ktoré sú anomáliami) na viac období dopredu je predpoveď spojená s väčšou neurčitosťou, ktorú treba zahrnúť do výpočtu predikčných intervalov.

Citlivosť algoritmu na anomálie je možné znížiť rozširovaním predikčných algoritmov (násobením určitou konštantou). Rozšírením predikčných intervalov dosiahneme väčšiu toleranciu algoritmu na extrémne hodnoty (ktoré môžu a nemusia byť anomáliami).

4.3.3 Detekcia anomálií na základe minimálneho a maximálneho toku

Zjednodušená verzia predchádzajúceho algoritmu berie do úvahy pri posudzovaní anomálie namiesto predikčných intervalov minimum a maximum historických pozorovaní. Ak je pozorovanie menšie ako minimum alebo väčšie ako maximum historických pozorovaní, bude nové pozorovanie označená ako stav 1, možná anomália. Ak je aj v nasledujúcom období pozorovanie mimo intervalu ohraničeného minimom a maximom, je hodnota označená ako anomália.

Aj v tomto algoritme sú anomálie vyraďované z dát, z ktorých sa počíta minimum a maximum. Stavy č. 1, možné anomálie sú zahrnuté do historických pozorovaní ak za nimi nenasledujú anomálie. Citlivosť tohto algoritmu je možné nastavovať zväčšovaním intervalu, na základe ktorého sa posudzuje anomália.

Výhodou tohto algoritmu je v porovnaní s predchádzajúcimi dvoma algoritmami výpočtová a pamäťová náročnosť. Na rozdiel od predchádzajúcich algoritmov nie je nutné nič počítať, algoritmus si musí uchovávať len minimum a maximum historických pozorovaní. Tento algoritmus nevie zachytiť trendy a ani sezónnosť.

4.3.4 Dvojkroková detekcia pomocou smerodajnej odchýlky

Pre experimentálne účely sme navrhli a implementovali algoritmus s dvomi stavmi využívajúci smerodajnú odchýlku. Od tohto algoritmu si sľubujeme väčšiu odolnosť proti náhodným zvýšeniam a zníženiam novej hodnoty, ktorá nie je anomáliou. Algoritmus pri prekročení násobku smerodajnej odchýlky prejde do stavu 1, možná anomália. Ak nadmerná hodnota pokračuje aj v ďalšom období, bude to vyhodnotené ako anomália.

4.3.5 Kombinovanie algoritmov pre detekciu anomálií

Výstupom detekcie anomálií je binárny stav – jedná sa o anomáliu alebo sa nejedná o anomáliu. Pre počítanie tohto stavu je možné použiť kombináciu vyššie uvedených algoritmov, nemusíme sa nutne rozhodnúť pre jediný algoritmus. Môžeme skombinovať algoritmus detekujúci anomálie pomocou predikčných intervalov spolu s algoritmom detekujúcim anomálie na základe smerodajnej odchýlky.

Anomália bude detekovaná, ak jeden z algoritmov detekuje anomáliu. Tým dosiahneme algoritmus citlivý na sezónnosť a trendy (pomocou predikčných intervalov). Algoritmus bude zároveň citlivý aj príliš vysoké vychýlenie od priemernej hodnoty, ktoré by inak nebolo detekované pomocou predikčných intervalov. Nevýhodou je zvýšená výpočtová náročnosť. Ďalšou nevýhodou bude pravdepodobne zvýšený počet falošných poplachov, čiže detekcií anomálií, ktoré nie sú anomáliou. Stačí, ak jeden model detekuje anomáliu, ktorá nie je anomáliou a tento algoritmus bude tiež detekovať ako anomáliu falošnú hodnotu.

Algoritmy pre detekovanie anomálií je nutné kombinovať s uvážením. Pomocou vhodných kombinácií je možné zvýšiť citlivosť, môže ale viesť k väčšiemu počtu falošných poplachov.

4.3.6 Problematika označovania skutočných anomálií

V predchádzajúcom texte sme sa zmienili, že dvojstavový algoritmus využívajúci predikčné intervaly nezahŕňa do výpočtu predikčných intervalov pozorovania, ktoré boli vyhodnotené ako anomálie. Ak by ich zahŕňal, postupom času by dochádzalo k znižovaniu citlivosti na detekciu anomálií. Tento prístup má podstatnú nevýhodu – ak algoritmus nesprávne detekuje anomáliu (false positive), v budúcnosti bude príliš citlivý a podobné pozorovania bude opäť detekovať ako anomálie (pretože pozorovanie detekované ako false positive nebolo zahrnuté do dát, z ktorých sa počítal predikčný interval). Tento problém nie je možné vyriešiť bez interakcie užívateľa. Užívateľ má pri použití tohto algoritmu 2 možnosti:

1. Nastaviť pri spustení detekcií správne parameter citlivosti algoritmu. Ak je správne nastavený tento parameter, algoritmus detekuje minimum falošných anomálií (false positive) a nedochádza k zvyšovaniu citlivosti algoritmu. Správne nastavenie citlivosti vyžaduje skúsenosť užívateľa s algoritmom.

2. Nastaviť parameter citlivosti na maximum a označovať falošné anomálie (false positive). Nastavením citlivosti na maximum dosiahne užívateľ vysoký počet falošných anomálií. Následne ich bude označovať ako nesprávne detekované anomálie a algoritmus ich začne zahrňovať do pozorovaní, z ktorých sa počítajú empirické predikčné intervaly. Tým dôjde k rozšíreniu empirických predikčných intervalov a zároveň aj k zníženiu falošne detekovaných anomálií. Po určitom čase dôjde týmto postupom k nastaveniu správnej citlivosti algoritmu. Pri tomto postupe je nutné udržiavať veľký počet pozorovaní, z ktorých sa počítajú empirické predikčné intervaly. To môže byť pamäťovo náročné.

Riešením by tiež mohlo byť postupné znižovanie parametru citlivosti algoritmu po každej nesprávne detekovanej falošnej anomálii. Po určitom čase by sme dosiahli rovnakého výsledku ako v možnosti č. 1 a nebolo by nutné uchovávať veľký počet historických pozorovaní.

Kapitola 5

Vyhodnotenie experimentov a diskusia

Cieľom tejto kapitoly je experimentálne overenie doteraz popisovaných algoritmov. Ako už bolo ukázané v kapitole 3, charakteristiky časových radov dátových tokov na sieťach sa môžu diametrálne líšiť. Preto budeme robiť experimenty na veľkom počte rôznych časových radov. Všetky experimenty boli simulované v štatistickom prostredí R[1] a spúšťané v nami implementovanom simulátore predikcií popísanom v prílohe H.

5.1 Automatická identifikácia AR modelu

V kapitole 4.1.4 sme sa zaoberali problematikou voľby rádu autoregresného modelu. Pre analýzu kvality predikcií popísaných algoritmov sme vytvorili experiment, ktorý na celkovom množstve 28 dátových sád simuloval predikcie. Cieľom tohto experimentu je porovnanie a vybratie najlepšej metódy pre automatickú identifikáciu autoregresného modelu.

Experimenty sme spúšťali s parametrami: predikcie boli počítané na 1, 3, 5 a 7 období dopredu, algoritmus sa učil na 40 a 80 historických pozorovaniach. Pre každú kombináciu týchto parametrov bol spustený samostatný experiment. V simuláciách boli použité pre predikovanie dátového toku nasledujúce predikčné modely:

- AR model s automatickou identifikáciou AR rádu pomocou PACF funkcie (v grafoch označený ako *pacf.ar.identificator*), pričom ako maximálna veľkosť AR rádu bola nastavená na 20. Mohlo byť teda pridaných maximálne 20 autoregresných členov, pričom najväčší mal rad 20.
- Ar model s automatickou voľbou AR rádu pomocou postupného pridávania AR členov (*automatic.ar.identificator*). Maximálny počet autoregresných členov bol opäť nastavený na 20.
- Ar model s autoregresnými členmi 1 až k (bf AR 1-k), k nadobúdalo hodnoty 1, 5, 10 15, 20.
- Algoritmus vyberajúci najlepší predikčný model na základe funkcie RMSE (*RMSE selector*) s obmedzeným počtom historických pozorovaní na 40 a 80, prípadne s variantou bez obmedzenia počtu historických pozorovaní (*RMSE selector without time horizont restriction*)
- Vážený priemer všetkých dostupných predikčných modelov (*All models average*), pričom ako hodnotiaca funkcia bola opäť použitá RMSE.
- Vážený priemer 3 najlepších modelov (*3 best models average*) s hodnotiacou funkciou RMSE.

1. http://www.r-project.org/

Ako hodnotiace kritérium sme zvolili metriku RMSE benchmark popísanú v kapitole 4.1.3. V grafe 5.1 uvádzame histogram metriky RMSE benchmark pre každý predikčný model. Pri výpočte metriky RMSE benchmark bol ako východzí model použitý autoregresný model s autoregresným členom rádu 1 (model *AR(1)*), pretože predstavuje primitívnu ľahko implementovateľnú predikčnú techniku – predpoveď zakladáme len na poslednom pozorovaní. V histograme sme zobrazili len hodnoty koeficientu RMSE benchmark väčšie ako -1 (100% zhoršenie oproti primitívnemu modelu). Modely, ktoré majú horšie hodnoty nemá význam ďalej posudzovať.

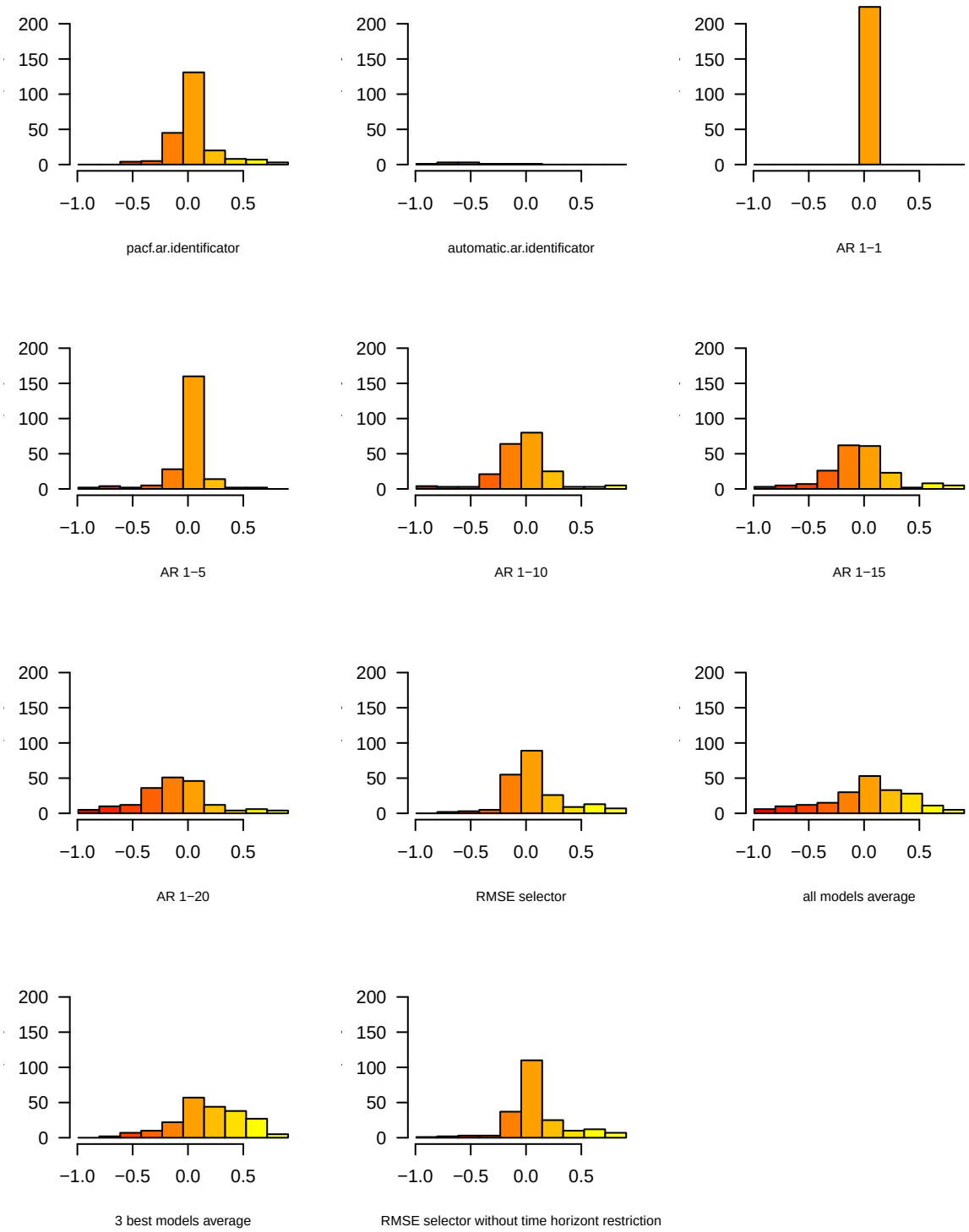
5.1.1 Vyhodnotenie experimentu

- Algoritmus *automatic.ar.identificator* nie je vhodný pre akékoľvek predikcie. Je to spôsobené problémom s pridávaním autoregresných členov a následným vyhodnocovaním modelov. Tento problém sme bližšie popísali v kapitole 4.1.4. 214 z 224 predikcií pomocou tejto techniky bolo o viac ako 100% horších ako predikcie pomocou primitívneho modelu.
- Algoritmus *pacf.ar.identificator* dosahoval dobré výsledky. V 94 z 224 prípadoch boli výsledky horšie primitívna technika, z toho 79 predikcií bolo horších o 0 až 20% ako primitívna technika. Pri 42 prípadoch z týchto predikcií bolo zhoršenie len o 5% oproti primitívnej technike. To považujeme za veľmi dobrý výsledok, počet predpovedí s nižšou kvalitou je menší ako počet predpovedí s väčšou kvalitou. Horšie predpovede sú z približne polovice len o 5% horšie ako predpovede pomocou primitívnej techniky.
- Algoritmy *RMSE selector*, *All models average*, *3 best models average* a *RMSE selector without time horizont restriction* stoja tiež za povšimnutie. Sú použiteľné ako alternatíva k technike *pacf.ar.identificator*. Ich nevýhoda je nutnosť vytvárania predikcií pomocou veľkého množstva predikčných modelov, z ktorých bude následne vybraná najlepšia predikcia. Naproti tomu metóda *pacf.ar.identificator* identifikuje najlepší model, ktorý je možné následne používať po niekoľko desiatok predikcií (podľa toho, ako sa mení charakteristika dát).

Pre voľbu autoregresného modelu odporúčame používať metódu *pacf.ar.selector*.

5.1.2 Analýza citlivosti parametrov

Pri predikcii pomocou vyššie uvedených algoritmov je mimo iné dôležitá voľba parametru počtu historických pozorovaní, ktoré budú do predikcie zahrnuté. Pokiaľ bude tento parameter príliš nízky, algoritmus nebude schopný predikovať podľa určitých sezónností, prípadne nebude schopný zachytiť určité trendy. Napríklad pri nastavení parametru na hodnotu 30 pri predikovaní na dátach s granularitou 1 záznam za hodinu nebude algoritmus schopný zachytiť týždňovú sezónnosť. Aby sme v tomto prípade mohli zachytiť týždňovú sezónnosť, musíme tento parameter nastaviť aspoň na hodnotu 168 (24*7). Ak by sme nastavili tento parameter na príliš vysoké hodnoty (prípadne nekonečno – brali by sa do úvahy všetky historické pozorovania), algoritmus by bol príliš výkonovo a pamäťovo náročný. Vzhľadom k tomu, že algoritmus pri identifikácii modelu vytvára v niektorých prípadoch až stovky modelov, mohlo by to spôsobiť výrazné spomalenie predikcií modelov. Z tohto dôvodu je nutné nastaviť algoritmus na vhodnú hodnotu.



Obr. 5.1: RMSE benchmark algoritmov predpovedajúcich pomocou AR modelu

Pri voľbe vhodného algoritmus pre vytváranie predpovedí je mimo iné nutné, aby boli jeho výsledky stabilné aj pri zmene parametrov. Preto sme sa rozhodli vytvoriť experiment, ktorý porovná citlivosť algoritmov z predchádzajúceho experimentu na zmeny parametru dĺžky počtu pozorovaní braných do úvahy pri predikciách. Cieľom je overenie, či sa v závislosti na konfigurácii parametrov nemenia dramaticky výsledky algoritmov. V grafoch E.1 a E.2 v uvedených v prílohe uvádzame simulácie na dátovej sade uvedenej v predchádzajúcich experimentoch a parametrami 40 a 80 historických pozorovaní zahrnutých do predikcií. Z grafov je pozorovateľný rozdielny výsledok metriky RMSE Benchmark, tento rozdiel však považujeme za zanedbateľný. Naše závery o algoritmoch slúžiacich na identifikáciu AR modelu sú platné aj pri zmene parametru počtu historických pozorovaní. Považujeme teda algoritmy stabilné pri zmene tohto parametru.

Ďalším dôležitým parametrom pri vytváraní predikcií je voľba na aké obdobie dopredu sa má vytvárať predikcia. Nami uvedené modely totiž rozlišujú, na aké obdobie dopredu vytvárame predikcie. Našim cieľom je vytvorenie takého algoritmu, ktorý bude vedieť vytvárať kvalitné predpovede na ľubovoľne zvolený počet období dopredu. V experimente uvedenom v tejto kapitole sme vyberali najvhodnejší algoritmus na základe predikcií na 1, 3, 5 a 7 období dopredu. Jednalo sa teda o krátkodobé predpovede. V nasledujúcej analýze citlivosti overíme, či sa výsledky RMSE Benchmarku výrazne líšili pri porovnaní predpovedí na rôzne obdobia dopredu (nebudeme teda do jedného histogramu RMSE Benchmarku agregovať predpovede na rôzne obdobia dopredu).

V grafe E.3 uvádzame histogram RMSE Benchmarku pre predpovede na 1 obdobie dopredu, v grafe E.4 na 3 obdobia dopredu, v grafe E.5 na 5 období dopredu a v grafe E.4 na 7 období dopredu. Z grafov je opäť badateľné rozdiely vo výsledkoch RMSE Benchmarku. Tieto rozdiely opäť považujeme za nevýznamné a naše závery o algoritmoch identifikujúcich AR model považujeme platné aj na základe týchto experimentov. Algoritmy považujeme za stabilné aj pri zmene parametrov.

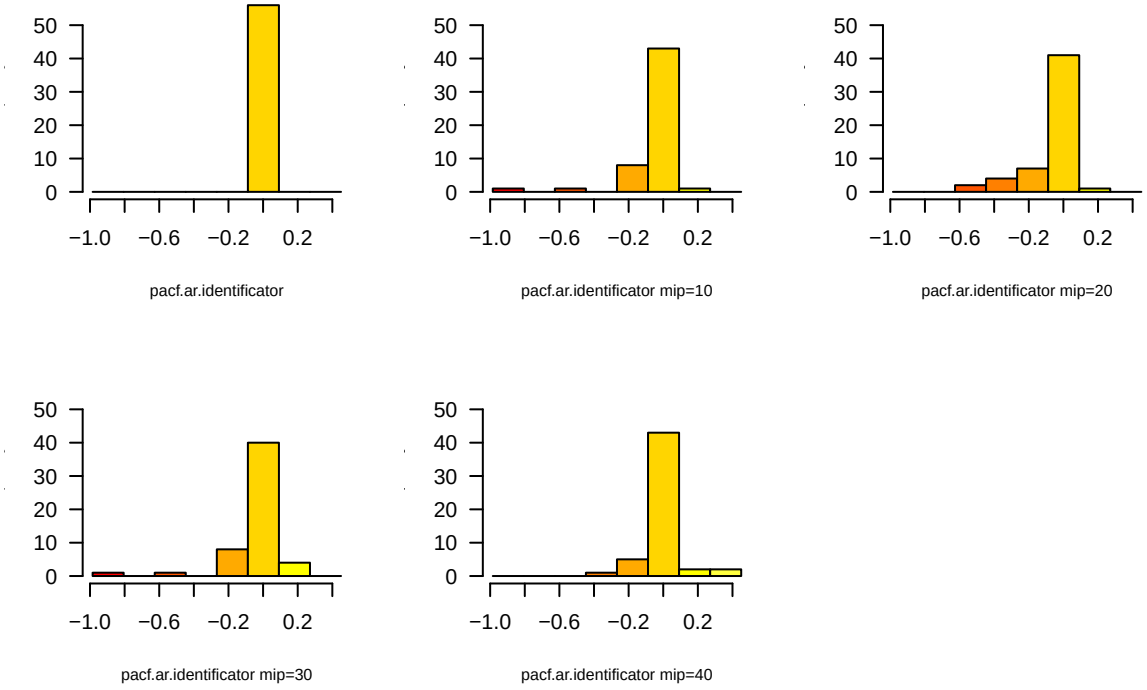
5.2 Voľba periódy pre detekciu AR modelu

V kapitole 4.1.4 sme uvádzali, že algoritmus detekujúci AR model nie je nutné spúšťať pred každou predikciou, ale stačí púšťať ho raz za určitú periódu. Ak dáta totiž nemenia svoju charakteristiku, môžeme očakávať, že algoritmus bude pre dáta detekovať rovnaký model. Tento predpoklad sme sa rozhodli experimentálne overiť. Vytvorili sme experiment, v ktorom sme na dátovej sade z CESNETu a simulovaných dátach simulovali predikovanie pomocou AR modelu detekovaného automaticky algoritmom využívajúcim PACF. Algoritmus detekujúci model sme spúšťali počas každej predikcie, každú 10., 20., 30. a 40. predikciu. Algoritmy boli porovnávané RMSE benchmarkom, pričom ako základný model bol použitý model detekovaný algoritmom spúšťaným po každej predikcii. V tomto experimente zisťujeme, či sa zmení výrazne kvalita predikcií ak sa zmení perióda automatického identifikovania modelu.

Pri simuláciách mali algoritmy pri predikciách k dispozícii 60 historických pozorovaní, predikcie boli vytvárané na 1 a 5 období dopredu.

5.2.1 Vyhodnotenie experimentu

Na základe grafu 5.2 môžeme konštatovať, že perióda, počas ktorej sa bude detekovať autoregresný model algoritmom využívajúcim PACF funkciu výrazne neovplyvňuje kvalitu pre-



Obr. 5.2: RMSE benchmark algoritmu detekujúceho automaticky AR model s rôznymi periódami detekcie modelu

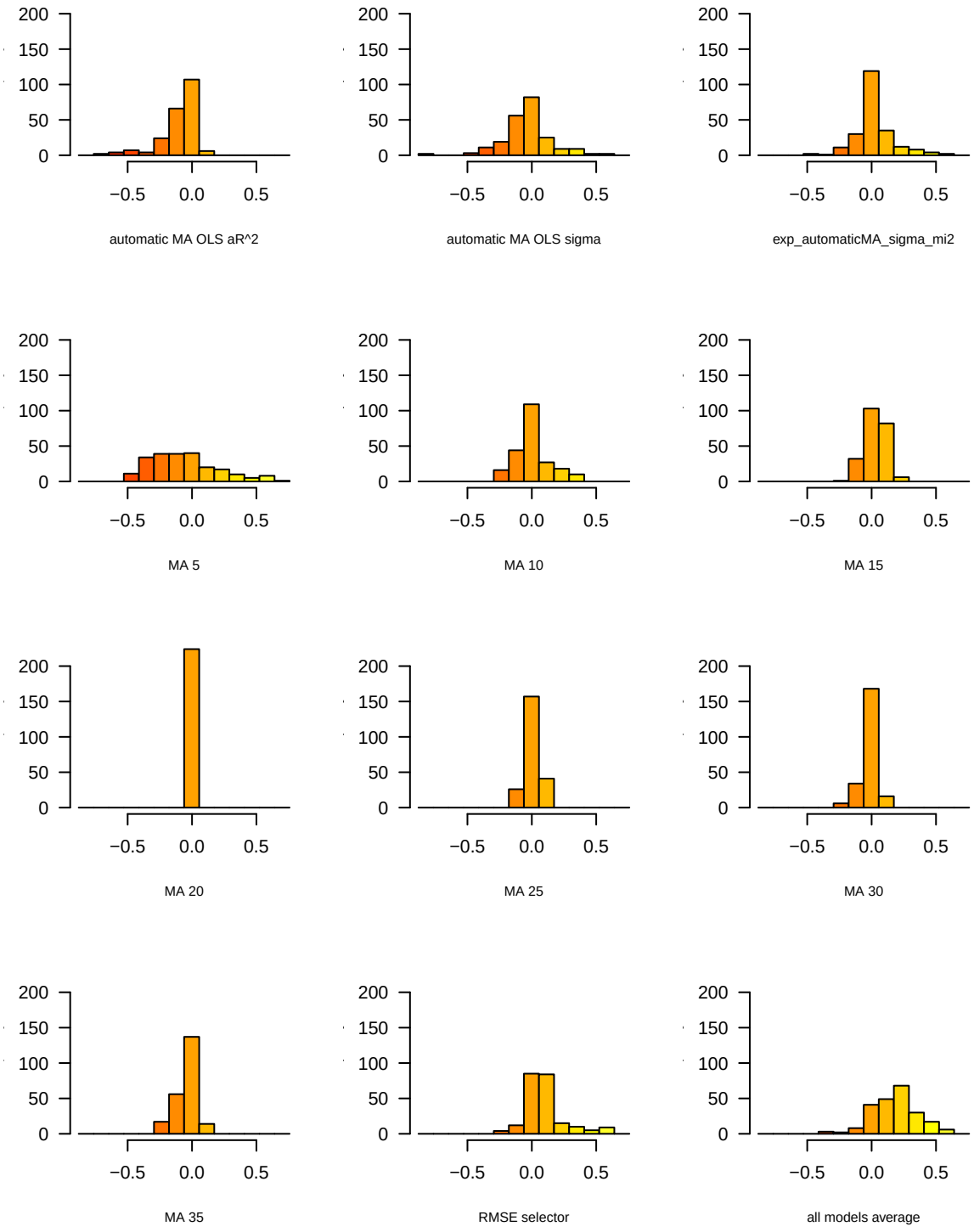
dikcií. Algoritmus s periódou 10 má síce 10 výsledkov horších o 0 až 20%, ale algoritmus s periódou 40 mal len 4 predikcie v tomto rozmedzí a 5 predikcií bolo dokonca lepších ako predikcie základného modelu. Konštatujeme, že na základe experimentu nie je možné povedať, že algoritmy s vyššou periódou majú horšie výsledky. Ich podstatou výhodou je menšia výpočtová náročnosť.

5.3 Algoritmy identifikujúce dĺžku okna MA modelov

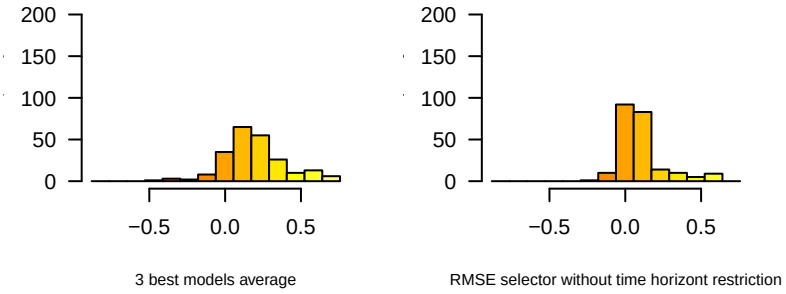
V kapitole 4.1.5 sme predstavili modely pre identifikáciu dĺžky okna modelu SMA – jednoduchého kĺzavého priemeru. V nasledujúcom experimente vyhodnotíme na dátovej sade z CESNETu a simulovaných dátach kvalitu jednotlivých predikcií generovaných pomocou týchto algoritmov. Opäť bude cieľom experimentu zvolenie algoritmu, ktorý bude dosahovať najlepšie výsledky. V experimente sme simulovali predikovanie pomocou nasledujúcich algoritmov a modelov:

- *automatic MA OLS sigma* je algoritmus automaticky detekujúci dĺžku SMA okna na postupným zvyšovaním okna, na porovnávanie veľkosti okna je použitý súčet štvorcov reziduí modelu. Model obsahuje trendovú konštantu.
- *automatic MA OLS aR^ 2* je rovnaký ako predchádzajúci algoritmus, ale na porovnávanie modelov je použitý koeficient determinácie.

- *exp_automaticMA_sigma_mi2* predstavuje algoritmus detekujúci dĺžku okna SMA modelu bez trendovej konštanty na základe súčtu štvorcov reziduí.
- SMA model s dĺžkou okna p (v grafe označovaný ako *MA p*), pričom p malo hodnoty 5, 10, 15, 20, 25, 30 a 35.
- Algoritmus vyberajúci najlepší model na základe hodnotiacej fukncie RMSE (*RMSE selector*) a jeho varianta bez obmedzenia počtu historických pozorovaní (*RMSE selector without time horizont restriction*)
- Vážený priemer všetkých dostupných predikčných modelov (*All models average*).
- Vážený priemer 3 najlepších modelov (*3 best models average*).



Obr. 5.3: RMSE benchmark algoritmu detekujúceho automaticky dĺžku okna SMA modelu, prvá časť



Obr. 5.4: RMSE benchmark algoritmu detekujúceho automaticky dĺžku okna SMA modelu, druhá časť

5.3.1 Vyhodnotenie experimetu

Graf 5.3 a 5.4 opäť predstavuje histogramy RMSE benchmarku jednotlivých simulovaných algoritmov. Ako primitívna technika pri výpočte RMSE benchmarku bol zvolený SMA model s dĺžkou okna 20.

- *automatic MA OLS aR^ 2* má len 5 predpovedí lepších ako primitívna predikčná technika. Ostatné predpovede sú horšie alebo rovnako dobré. Tento algoritmus považujeme za nevhodný, v drvivej väčšine prípadov je výhodnejšie používať primitívnu predikčnú techniku.
- *automatic MA OLS sigma* má tiež väčšinu predpovedí horšiu ako primitívna predikčná technika, považujeme ho taktiež za nevhodný pre akúkoľvek detekciu modelu.
- *exp_automaticMA_sigma_mi2* dosahoval v 133 zo 224 predpovedí lepšie výsledky ako primitívna predikčná technika. Táto technika má približne rovnako kvalitné predpovede ako primitívna predikčná technika.
- *All models average* – model vážených priemerov všetkých dostupných dosahoval dobré výsledky. V našom prípade sa jednalo o vážený priemer SMA modelov s dĺžkou okna 5, 10, 15, 20, 25, 30 a 35, pričom ako váha priemeru bola použitá normalizovaná funkcia RMSE. Tento model nie je výpočtovo náročnejší ako primitívny model (namiesto jedného SMA modelu sa počíta 7 SMA modelov) a v porovnaní s primitívnym modelom dosahoval v 196 z 224 simulácií lepšie výsledky, pričom v 13 prípadoch bolo zhoršenie do 5% percent.
- *RMSE selector* – algoritmus vyberajúci najlepší predikčný model na základe hodnotiacej funkcie RMSE dosahoval uspokojivé výsledky. Jeho výsledky boli v 169 z 224 prípadov lepšie ako výsledky primitívneho modelu a v 35 prípadoch bolo zhoršenie do 5% oproti primitívnemu modelu.

Z našich algoritmov automaticky detekujúcich AR model nie je podľa nás ani jeden vhodný pre praktické použitie. Lepšie alebo porovnateľne dobré výsledky sme dosiahli pri

použití primitívnej techniky, čiže jednoduchého kĺzavého priemeru s dĺžkou okna 20. Oproti technikám automaticky detekujúcim model je tento model výpočtovo oveľa jednoduchší.

Dobré výsledky dosahoval model vážených priemerov všetkých dostupných algoritmov (*All models average*) a algoritmus vyberajúci zo všetkých dostupných modelov najlepší (*RMSE selector*). Ich výpočtová náročnosť je podobná, vážený priemer všetkých dostupných modelov ale dosahoval lepšie predikčné výsledky. Jeho výhoda spočíva v tom, že do výpočtu zahrňuje viacero modelov s rôznymi váhami, zatiaľ čo RMSE selector sa obmedzuje len na jeden model, v ktorom majú všetky pozorovania rovnakú váhu (pretože sa jedná o techniku SMA). Pre predikcie pomocou modelu MA odporúčame používať algoritmus vážených priemerov viacerých modelov.

5.4 Algoritmy pre detekciu umelých premenných

Cieľom tohto experimentu je porovnanie algoritmov pre automatickú identifikáciu umelých premenných popísaných v kapitole 4.1.6 a výber algoritmu dosahujúceho najlepšie výsledky. Experimenty budeme opäť spúšťať na vybraných časových radoch z prevádzky siete CESNET2 a na simulovaných časových radoch. Algoritmy pre identifikáciu umelých premenných mali nastavené rovnaké parametre pre dĺžku maximálnej sezónnosti umelých premenných. Ako východzí model pre identifikáciu umelých premenných bol použitý AR model, ktorý bol identifikovaný pomocou algoritmu identifikujúceho na základe PACF funkcie. Algoritmy pre identifikáciu modelu boli vzhľadom k časovej náročnosti spúšťané každých 20 predikcií. Algoritmy boli hodnotené RMSE Benchmarkom, ako primitívny model boli vybraný autoregresný model *AR(1)*. V experimente sme simulovali predikovanie pomocou nasledujúcich algoritmov a modelov:

- Autoregresný model AR(k) označený v grafe ako *AR 1-k*
- AR model automaticky identifikovaný pomocou algoritmu využívajúceho funkciu pacf, označený ako *pacf.ar.identificator*
- AR model identifikovaný pomocou funkcie PACF s umelými premennými identifikovanými pomocou algoritmu pridávajúceho hromadne umelé premenné, v grafe označený ako *experimantal av 1a*
- AR model identifikovaný pomocou funkcie PACF s umelými premennými identifikovanými algoritmom pridávajúcim postupne umelé premenné, umelé premenné boli pridávané v opačnom poradí (najprv s dĺžkou sezónnosti N), v grafe je algoritmus označený ako *experimantal av 1b*
- AR model identifikovaný pomocou funkcie PACF s umelými premennými identifikovanými algoritmom pridávajúcim postupne umelé premenné, v grafe je algoritmus označený ako *experimantal av 1c*
- Algoritmus vyberajúci najlepší model pre predikcie zo všetkých dostupných modelov na základe hodnotiacej funkcie RMSE, v grafe označený ako *RMSE selector*

5.4.1 Vyhodnotenie experimentu

V grafe 5.5 uvádzame výsledky experimentu s algoritmami identifikujúcimi umelé premenné. Graf zobrazuje len výsledky RMSE Benchmarku, v ktorých dosiahli algoritmy zhor-

šenie najviac 100% oproti primitívnemu modelu a lepšie (graf je teda obmedzený na hodnoty RMSE Benchmarku -1 a väčšie). Algoritmy, ktoré dosahovali nižšie hodnoty nepovažujeme relevantné pre akékoľvek použitie.

- *experimantal av 1b* a *experimantal av 1c* dosahovali viac ako polovicu výsledkov horších ako 100%. Znamená to, že tieto algoritmy vedú k výraznému zhoršeniu predpovede. Algoritmy pridali príliš veľa umelých premenných, čo viedlo k nesprávne odhadnutým parametrom modelu (pomocou metódy OLS). Postupné pridávanie umelých premenných do modelu na základe tohto experimentu vyhodnocujeme ako nevhodné pre akékoľvek predikcie a v ďalšom texte sa týmito algoritmami nebudeme zaoberať.
- *experimantal av 1c* dosahoval podobné výsledky ako bf pacf.ar.identificator, teda AR model bez umelých premenných. Podrobnejšia analýza predpovedí ukázala, že v niekoľko málo prípadoch viedlo použitie modelu s umelými premennými k zlepšeniu premenných. Algoritmus častejšie viedol k horším predpovediam ako samotný automaticky identifikovaný AR model aj model *AR(1)*. Model s umelými premennými obsahoval príliš veľa umelých premenných. Použitie tohto algoritmu pre predikcie je teda sporné, je výpočtovo náročnejší ako samotný algoritmus identifikujúci AR model a výsledky sú približne rovnaké.
- *RMSE selector* dosahoval zaujímavé výsledky. V 62 zo 127 prípadov boli jeho výsledky lepšie ako výsledky primitívneho modelu, z toho v 20 prípadoch boli výsledky lepšie o viac ako 20%. Z 67 zo 127 prípadov zhoršení bolo 45 zhoršení výsledkov v rozmedzí do 2% a 57 do 5%. To považujeme za veľmi nízke zhoršenie, prisudzujeme to počiatočnému behu algoritmu, keď je pre výpočet hodnotiacej funkcie RMSE dostupných príliš málo predikčných chýb. Algoritmus síce vo viac ako v polovici prípadov predpovedal horšie ako primitívny model, ale toto zhoršenie bolo veľmi nízke. V prípadoch, v ktorých predpovedal lepšie ako primitívny algoritmus bolo zlepšenie podstatné.

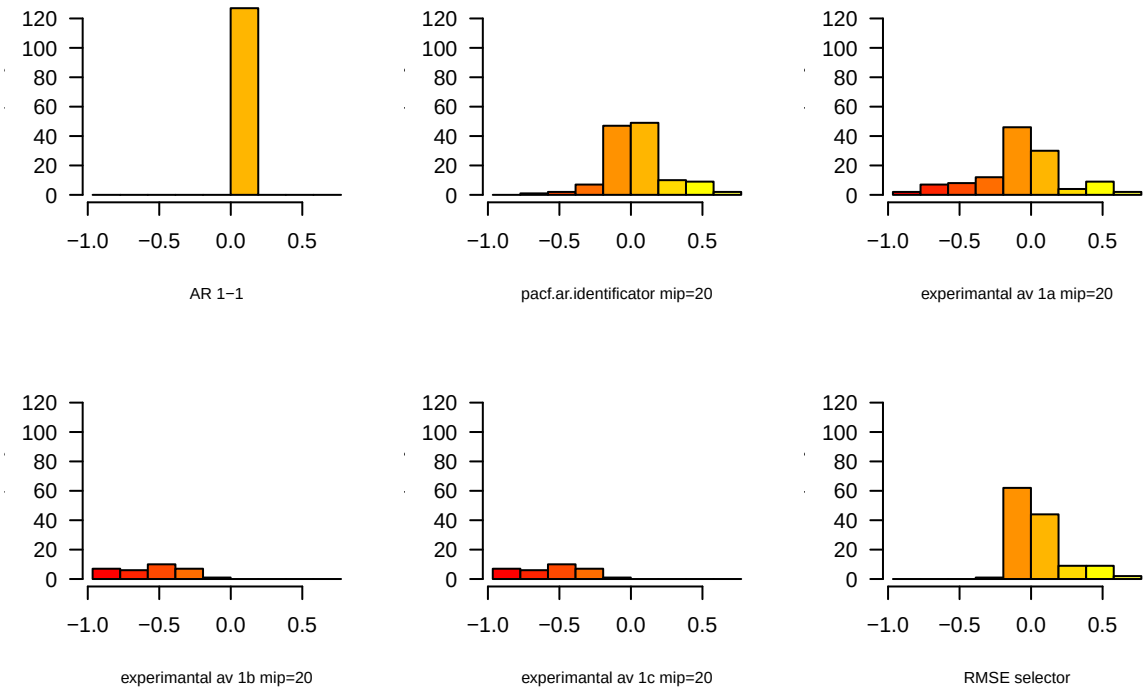
Pri použití modelov s umelými premennými odporúčame používať algoritmus *RMSE selector*, jeho výsledky považujeme lepšie ako výsledky primitívneho modelu.

Neodporúčame používať algoritmy *experimental av 1b* a *experimental av 1c*. Samotný algoritmus *experimantal av 1c* tiež nie je vhodný pre predikcie. Je možné použiť ho ako jeden z modelov v algoritme vyberajúcom zo všetkých dostupných modelov (napr. RMSE Selector). V takom prípade nevadí, že algoritmus na niektorých typoch dát identifikuje príliš komplexné modely. V takýchto prípadoch algoritmus RMSE selector vyberie iné modely (napr. *AR(1)*, *MA*, prípadne automaticky identifikovaný model AR pre predikcie.

5.5 Algoritmy pre elimináciu umelých premenných lineárneho modelu

V kapitole 4.1.7 boli popísané algoritmy, ktoré eliminujú automaticky detekované premenné lineárneho modelu. Experimentálne sme overili pomocou simulácie predikcií, ktorý z algoritmov dosahuje najlepšie výsledky. Primitívnym modelom RMSE Benchmarku bol v tomto prípade model *AR(1)*. Algoritmy pre identifikáciu modelu boli spúšťané každú tretiu predikciu. V nasledujúcom výpise uvádzame popis algoritmov, pomocou ktorých sme simulovali predikovanie a uvádzame tiež značenie týchto algoritmov v grafe.

- Autoregresný model AR(k) označený v grafe ako *AR 1-k*

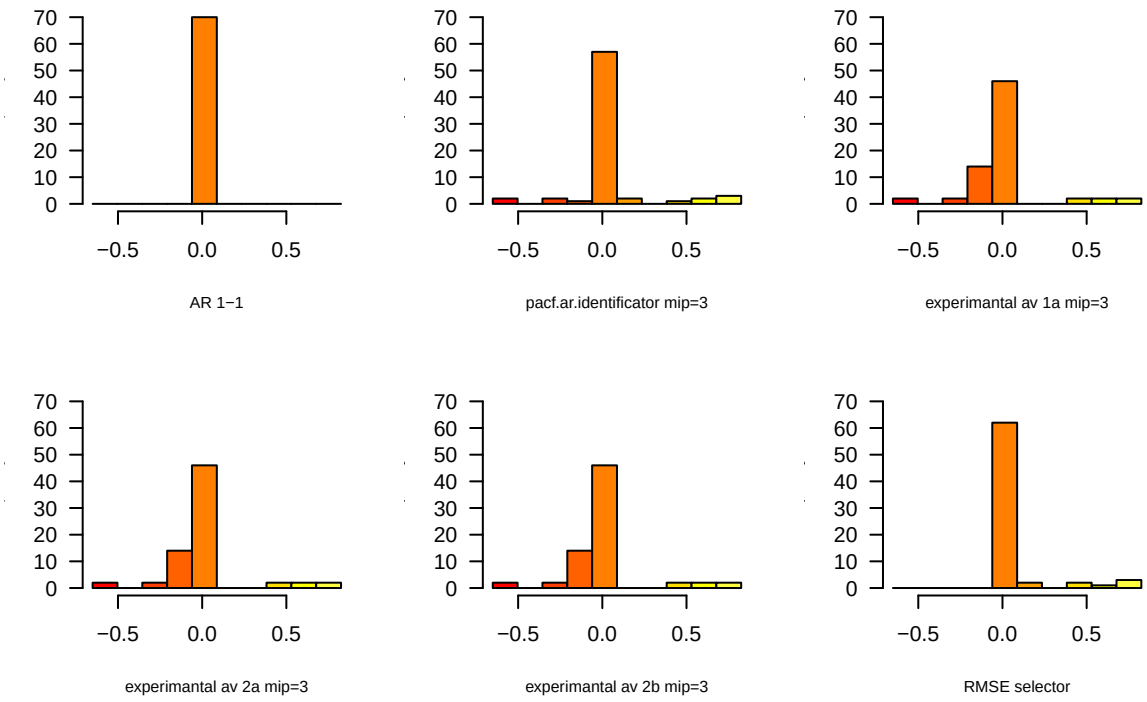


Obr. 5.5: RMSE benchmark algoritmu detekujúceho automaticky umelé premenné lineárneho modelu

- AR model automaticky identifikovaný pomocou algoritmu využívajúceho funkciu pacf, označený ako bf pacf.ar.identificator
- AR model identifikovaný pomocou funkcie PACF s umelými premennými identifikovanými pomocou algoritmu pridávajúceho hromadne umelé premenné, v grafe označený ako *experimantal av 1a*
- Algoritmus identifikoval model pomocou *experimantal av 1a* a následne eliminoval identifikované premenné pomocou t-testu *experimantal av 2a*
- Algoritmus identifikoval model pomocou *experimantal av 1a* a cyklicky eliminujúci identifikované premenné model na základe koeficientu $\hat{R}^2$ *experimantal av 2b*
- Algoritmus vyberajúci najlepší model pre predikcie zo všetkých dostupných modelov na základe hodnotiacej funkcie RMSE, v grafe označený ako *RMSE selector*

Graf 5.6 zobrazuje výsledky experimentu. Algoritmy *experimantal av 2a* a *experimantal av 2b* nedosahovali žiadne zlepšenie oproti algoritmu *experimantal av 2a*. Bolo to spôsobené tým, že neeliminovali žiadne premenné z modelu a tým pádom predpovedali pomocou rovnakého modelu, ako algoritmus *experimantal av 1a*. Experiment nepotvrdil, že by tieto algoritmy mali nejaký význam pre správne identifikovanie modelu.

Pre praktické použitie stále môžu mať tieto algoritmy význam. Ak budú predikčné algoritmy identifikujúce model bežať na dátach, na ktorých budú identifikovať kolineárne



Obr. 5.6: RMSE benchmark algoritmu eleminujúceho automaticky identifikované premenné z lineárneho modelu

premenné modelu, bude nutné spustiť tieto algoritmy. To v tomto experimente nebol náš prípad.

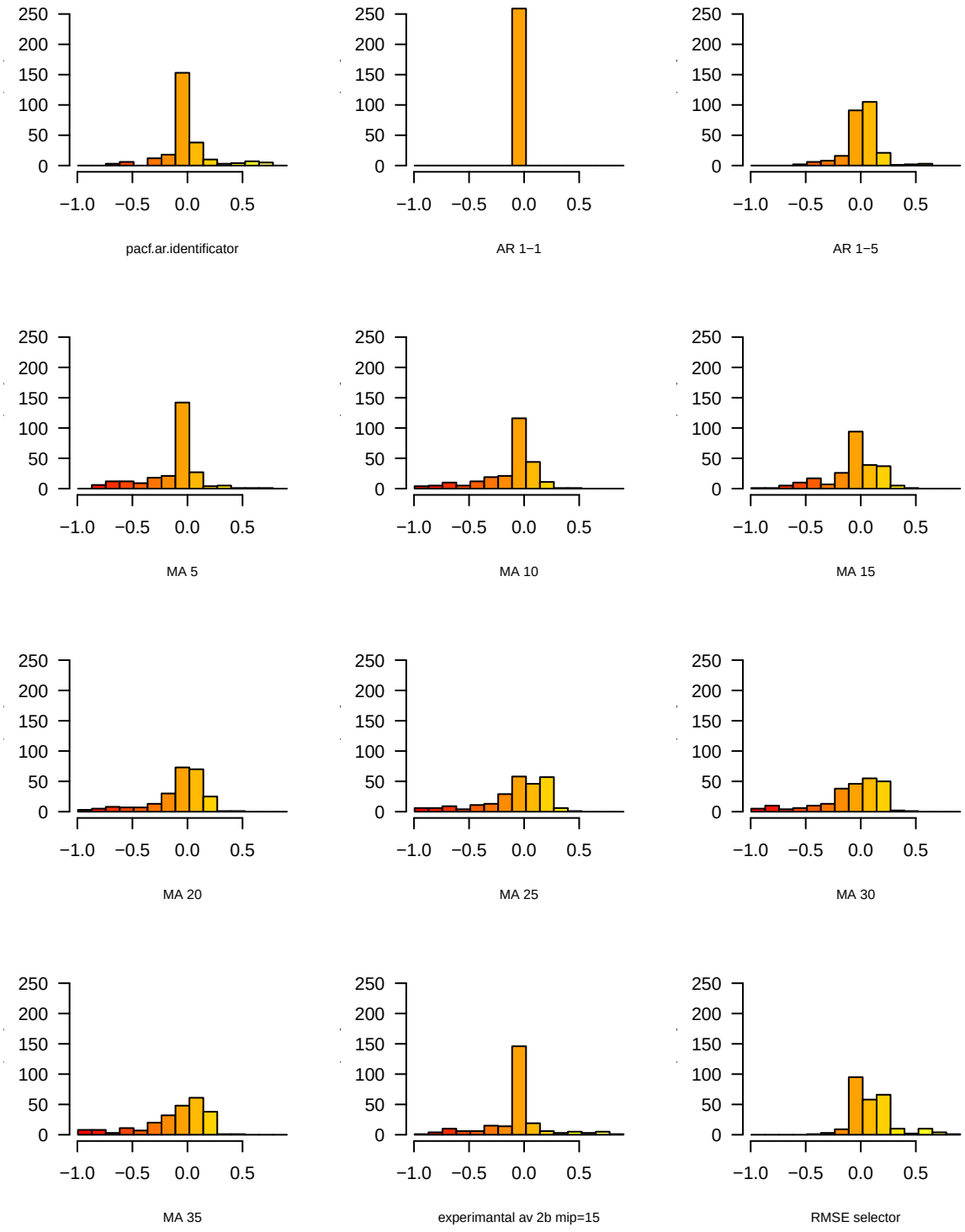
5.6 Algoritmy pre výber najlepšieho modelu

V nasledujúcom experimente budeme porovnávať všetky vyššie uvedené algoritmy pre vytváranie predikcií na základe lineárnych modelov a kĺzavých priemerov. Ďalej budeme v tomto experimente predikovať pomocou algoritmov vyberajúcich zo všetkých dostupných predikčných algoritmov. Cieľom experimentu je vybrať najvhodnejší algoritmus pre vytváranie predpovedí dátových tokov. Modely sme opäť porovnávali na základe nami vytvoreného kritéria RMSE Benchmark. Ako primitívny model slúžil model *AR(1)*. Predikcie boli vytvárané na dátovej sade CESNET2 a simulovaných dátach. V nasledujúcom texte popíšeme jednotlivé algoritmy, na základe ktorých sme simulovali predikcie.

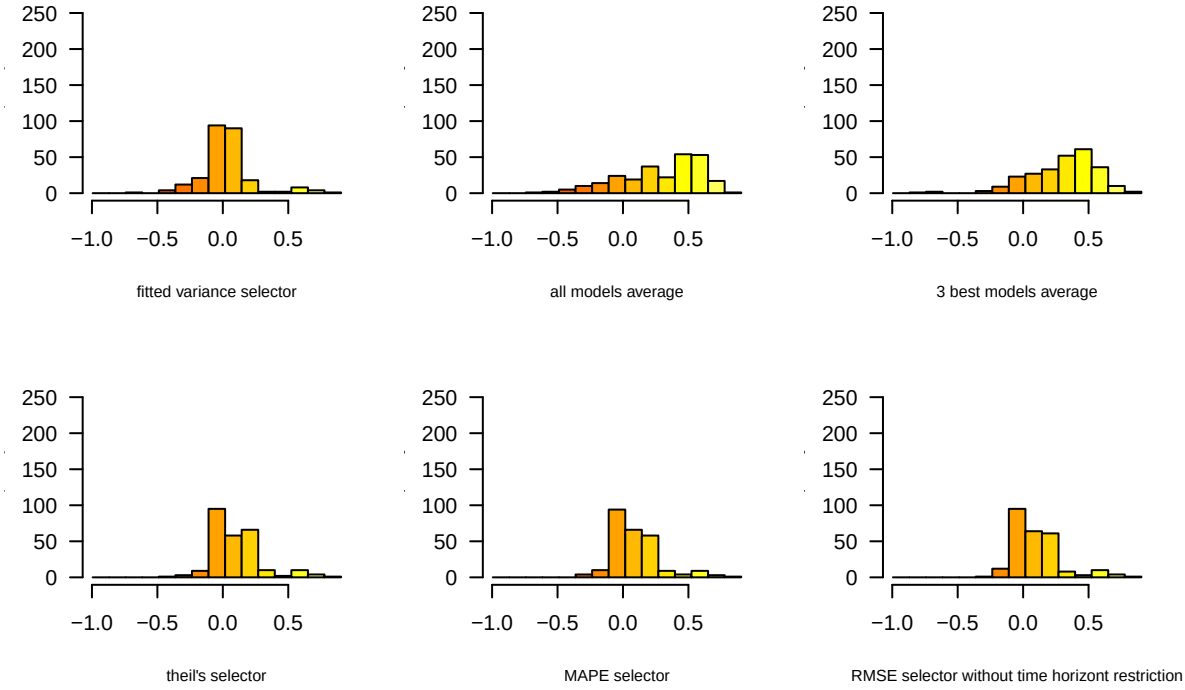
- Autoregresný model AR(1) označený v grafe ako *AR 1-1* a model AR(5) označený ako *AR 1-5*
- AR model automaticky identifikovaný pomocou algoritmu využívajúceho funkciu pacf, označený ako *pacf.ar.identificator*
- AR model identifikovaný pomocou funkcie PACF s umelými premennými identifikovanými pomocou algoritmu pridávajúceho hromadne umelé premenné, v grafe

označený ako *experimantal av 2b*

- Algoritmus vyberajúci najlepší model pre predikcie zo všetkých dostupných modelov na základe hodnotiacej funkcie RMSE (označený ako *RMSE selector*), na základe hodnotiacej funkcie MAPE (označený ako *MAPE selector*), na základe hodnotiacej funkcie Theil's U (označený ako *theil's U selector*), hodnotiacej funkcie veľkosti reziduí odhadnutého modelu (označeného *fitted variance selector*) a hodnotiacej funkcie RMSE bez obmedzenia počtu historických pozorovaní – algoritmus má nekonečnú pamäť (označený v grafe ako *RMSE without time horizont restriction*)
- SMA model s dĺžkou okna p (v grafe označovaný ako *MA p*), pričom p malo hodnoty 5, 10, 15, 20, 25, 30 a 35.
- Vážený priemer všetkých dostupných predikčných modelov (*All models average*)
- Vážený priemer 3 najlepších modelov (*3 best models average*)



Obr. 5.7: RMSE benchmark algoritmov pre výber najlepšieho modelu, prvá časť



Obr. 5.8: RMSE benchmark algoritmov pre výber najlepšieho modelu, druhá časť

5.6.1 Vyhodnotenie experimentu

O žiadnom zo samostatne použitých lineárnych modelov AR a MA s vopred stanovenými modelmi (nie automaticky identifikovanými) sa nedá povedať, že by dosahoval jednoznačne lepšie výsledky na všetkých simulovaných dátových sadách. V určitých prípadoch predikovali lineárne modely lepšie ako model *AR(1)* – AR model rádu 1, nastalo ale veľa prípadov, v ktorých boli predikcie významne horšie. Tento fakt si vysvetľujeme tým, že v niektorých prípadoch nebol použitý na dátovú sadu správny model. Predikovali sme pomocou nesprávneho modelu, čo viedlo k zlým predpovediam. Z modelov, ktoré sme použili v simulácii nie je možné vybrať jeden jednoduchý model použiteľný na všetky typy dát. Všetky modely dosahovali v niektorých prípadoch lepšie výsledky ako primitívny model. Nie je teda možné jednoznačne povedať, že niektorý z uvedených modelov by bol vo všetkých prípadoch horší ako primitívny model. V nasledujúcom texte porovnáme jednotlivé algoritmy:

- Predikčné techniky založené na automatickej identifikácii AR modelu (*pacf.ar.identificator*) a AR modelu s umelými premennými (*experimantal av 2b*) dosahovali v niektorých prípadoch lepšie výsledky ako primitívny model, v niektorých prípadoch ale dosahovali horšie výsledky. Opäť si to vysvetľujeme nesprávne aplikovaným modelom na tento typ dát.
- Algoritmy vyberajúce predikcie zo všetkých dostupných modelov *MAPE selector*, *theil's U selector* a *RMSE selector* fungovali na rovnakom princípe – na základe hodno-

tiacej funkcie spočítanej z predikčných chýb vybrali jeden model a pomocou neho vytvárali ďalšie predikcie. Rozdiel pri týchto algoritmoch je len v hodnotiacej funkcii. Nie je možné povedať, ktorá z hodnotiacich funkcií je najlepšia. Všetky vychádzajú z predikčných chýb. Líšia sa tým, ako veľmi penalizujú veľké predikčné chyby oproti malým predikčných chybám. Z výsledkov experimentu môžeme konštatovať, že výsledky týchto algoritmov sú veľmi podobné a nie je možné jednoznačne rozhodnúť, ktorý algoritmus je najlepší. Výsledky algoritmov sú tiež ovplyvnené tým, že ako kritérium je použitý RMSE Benchmark vychádzajúci práve z kritéria RMSE. Výsledky algoritmu *RMSE selector* sú vo výsledkoch našeho experimentu mierne zvýhodnené oproti výsledkom *MAPE selector* a *theil's U selector*, pretože v algoritme aj vo výslednej hodnotiacej funkcii experimentu je použitá funkcia RMSE. Ak by sme použili pre hodnotenie experimentu funkciu MAPE, bol by zvýhodnený *MAPE selector*.

- Variant algoritmu *RMSE selector* bez obmedzenia počtu historických pozorovaní časovej rady, *RMSE without time horizont restriction* dosahoval pri predikciách podobné výsledky ako samotný *RMSE selector* (ktorý mal obmedzený počet historických pozorovaní, ktoré bral do úvahy). Na základe toho konštatujeme, že rozumná voľba obmedzenia počtu historických pozorovaní neznižuje kvalitu predikcií. Zámerne neuvádzame, čo si predstavujeme pod pojmom rozumná voľba počtu historických pozorovaní, pretože to závisí na viacerých faktoroch. Konkrétne sa jedná napríklad o granularitu dát a dĺžku sezónností, ktoré sa nachádzajú v dátach. Pre časové rady s granularitou 1 deň a mesačnou sezónnosťou by sme odporučili 60-90 historických pozorovaní. Pre časové rady s granularitou 1 hodina a týždňovú sezónnosť je nutné použiť 168 až 336 historických pozorovaní.

- Algoritmus označený ako *fitted variance selector* tiež vyberá z dostupných jednoduchých modelov, pre výber ale nepoužíva hodnotiacu funkciu založenú na predikčných chybách. Jeho hodnotiaca funkcia je založená na reziduách odhadu modelu. Pre použitie tohto algoritmu nie je nutné mať dostupné predikčné chyby, stačí len odhadnúť parametre každého modelu. To je výhodné najmä v prípade, keď nemáme dostatok pamäte pre ukladanie predikčných chýb. Tiež je to výhodné v prípade začiatku predikcie. Ak by sme chceli použiť algoritmus *RMSE selector* na začiatku predpovedania, budeme musieť počkať niekoľko pozorovaní, dokým nezískame prvé predikčné chyby. S *fitted variance selector* nám stačí pre výber modelu odhadnúť parametre modelov, nepotrebujeme žiadne predikčné chyby. Tento algoritmus dosahoval horšie predikčné výsledky ako *RMSE selector*. Pri porovnávaní reziduí odhadu modelov sú vždy zvýhodnené modely s viacerými premennými oproti modelom s menším počtom premenných (z vlastností metódy najmenších štvorcov vyplýva, že pridanie premennej do modelu nemôže zhoršiť reziduá odhadnutého modelu). Pridanie premennej do modelu ale môže zhoršiť predikčné schopnosti modelu. Preto odporúčame v prípade dostatku predikčných chýb použiť radšej model *RMSE selector*. Ten nie je ovplyvnení počtom premenných modelov.

- *All models average* a *3 best models average* dosahovali veľmi zaujímavé výsledky. Tieto algoritmy pre výpočet váh váženého priemeru používali hodnotiacu funkciu RMSE (založenú na predikčných chybách). S tým sa spájajú nevýhody popísané v predchádzajúcom odstavci. Výsledky týchto modelov dosahovali lepšie predikčné kvality ako použitie jedného modelu algoritmom *RMSE selector*. Je to spôsobené tým, že do

výslednej predpovede bolo možné zahrnúť viacero jednoduchých modelov. Rozdiel medzi *All models average* a *3 best models average* považujeme za nepodstatný. Medzi týmito dvomi algoritmami nie je ani rozdiel vo výpočtovej náročnosti (oba algoritmy musia stále počítať predpovede na základe všetkých predikčných modelov a oba algoritmy musia uchovávať určitý počet predikčných chýb). Algoritmus *All models average* považujeme za univerzálnejší. Ak by boli predikcie aplikované na dátach, pri ktorých by bolo výhodné skombinovať 4 a viac jednoduchých modelov, tento algoritmus by dosahoval lepšie výsledky.

Pre predikcie neznámych časových radov odporúčame používať algoritmus *All models average*, pričom do modelov zahrnutých pre výpočet váženého priemeru by sme zahrnuli všetky vyššie uvedené jednoduché modely. Ak by sme pre predikcie nemali dostatok výpočtového výkonu, zahrnuli by sme len modely obsahujúce vopred definované modely – *AR(1)*, *AR(5)*, prípade až *AR(30)* podľa typov dát a *MA(5)* a vyššie, opäť podľa granularity dát. Pre začiatok predpovede, keď ešte nie je dostupný dostatočný počet predikčných chýb odporúčame použiť algoritmus *fitted variance selector*.

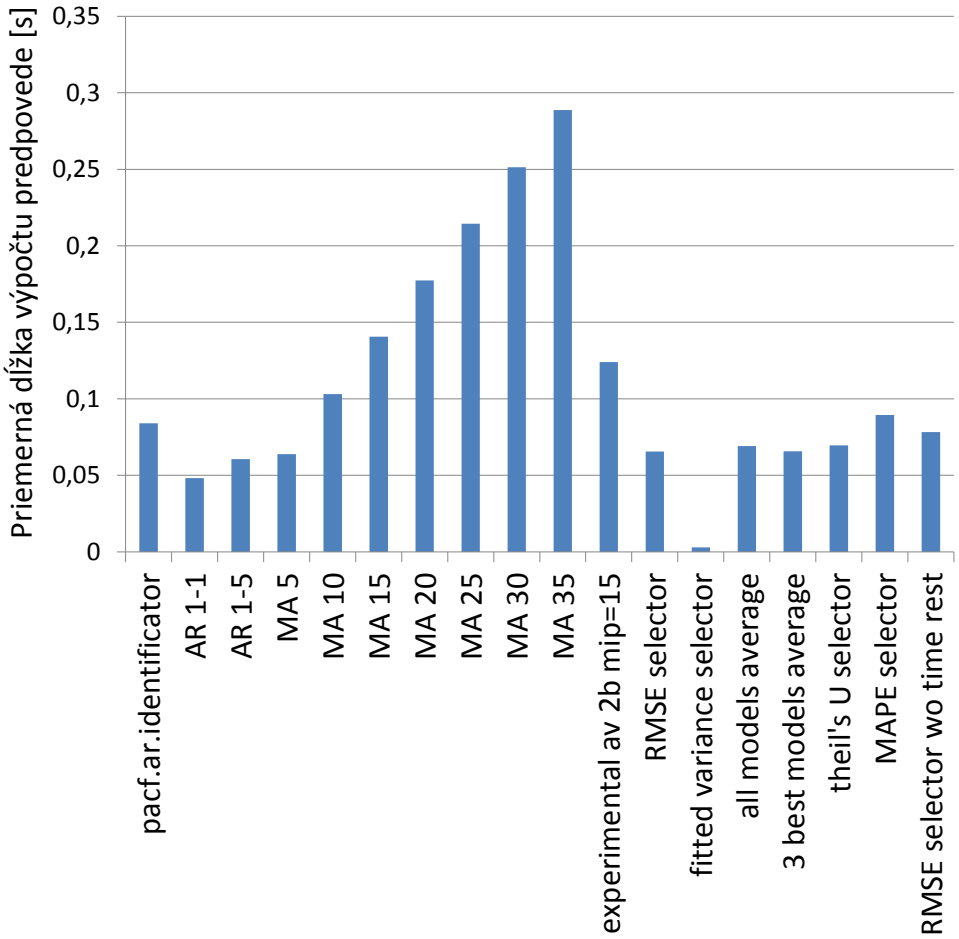
Výhodou algoritmov *RMSE variance selector* a *All models average* je fakt, že do modelov z ktorých sa vyberá najvhodnejší model je možné zahrnúť ďalšie zložitejšie modely, ako napríklad ARIMA, neurónové siete a podobne. Záleží na tom, ako veľa výpočtového výkonu máme a či máme vopred nejaké znalosti o časových radoch, ktoré chceme predikovať.

5.6.2 Porovnanie výkonnosti

Dostatočná rýchlosť vytvárania predikcií je jednou z našich priorít pri navrhovaní predikčných algoritmov. Preto sa teraz budeme venovať analýze výpočtovej náročnosti predikčných algoritmov. Zameriame sa na dĺžku trvania výpočtu jednej predpovede. Výsledky tejto analýzy boli zhromaždené pri predchádzajúcom experimente. Pri predikciách bolo použitých 80 historických pozorovaní. Značenie algoritmov odpovedá značeniu algoritmov z predchádzajúceho experimentu.

Graf 5.9 znázorňuje priemernú dĺžku trvania jednej predikcie. Za zmienku stoja nasledujúce výsledky algoritmov:

- AR(1) je z jednoduchých predikčných modelov je podľa očakávania najrýchlejší.
- AR(5) je približne 1,25x náročnejší kvôli vyššiemu počtu členov.
- *experimantal av 2b* a *pacf.ar.identificator* obsahujú taktiež model AR, pri ich výpočte dochádza každých 15 predikcií k identifikácii modelu. Ak by identifikácia modelu prebiehala častejšie, výpočtová náročnosť algoritmov by sa zvýšila.
- Výpočtová náročnosť modelov MA stúpa s počtom použitých členov v modeli.
- Algoritmy *RMSE selector*, *MAPE selector*, *theil's U selector* a *RMSE without time horizont restriction*, vyberajúce najlepší model podľa hodnotiacej funkcie, majú podobnú výpočtovú náročnosť.
- Algoritmy *All models average* a *3 best models average* kombinujúce modely pomocou váženého priemeru majú podobnú náročnosť ako algoritmy vyberajúce model na základe hodnotiacej funkcie.



Obr. 5.9: Porovnanie výpočtovej náročnosti predikčných algoritmov

- Algoritmus *fitted variance selector*, ktorý vyberá najlepší model podľa odhadu lineárneho modelu na historických dátach, má najnižšiu náročnosť. Tento algoritmus totiž nepočíta na rozdiel od ostatných algoritmov hodnotiacu funkciu, berie do úvahy veľkosť reziduí z odhadnutých modelov.

V náročnosti algoritmov vyberajúcich najlepší model zo všetkých dostupných algoritmov nie je zahrnutý čas výpočtu predikcií pomocou týchto algoritmov. Pre beh týchto algoritmov musia byť pri každej predikcii spustené všetky dostupné algoritmy. Čas uvedený v grafe 5.9 odpovedá len výpočtu samotnej procedúry výberu najlepšieho predikčného algoritmu, ak už máme spočítané predikcie pomocou dostupných algoritmov. Skutočný čas behu týchto algoritmov teda odpovedá súčtu dĺžky trvania výpočtu všetkých predikčných modelov a dĺžky trvania výpočtu algoritmu vyberajúceho najlepší model.

Pre praktické použitie je taktiež dôležitá pamäťová náročnosť algoritmov. Tou sa v tejto práci nebudeme zaoberať, pretože výsledky z nášho simulátora nemajú vypovedaciu hodnotu. Pamäťové štruktúry používané v našom simulátore obsahujú veľké množstvo údajov slúžiacich pre ladenie algoritmov. Nami namerané výsledky by neodpovedali realite. Pamä-

ťová náročnosť tiež bude závisieť na implementácii algoritmov pre násobenie matíc (použitom v odhade modelu pomocou metódy najmenších štvorcov). Odporúčame analyzovať pamäťovú náročnosť na finálnej implementácii algoritmov v programovacom jazyku, v ktorom budú predikčné algoritmy používané.

Na základe výsledkov z tejto analýzy a z predchádzajúceho experimentu zaoberajúceho sa kvalitou predikcií by sme zhodnotili naše závery následovne: Výpočtovo najrýchlejšie, ale najmenej presné predikcie poskytuje model *AR(1)*. Pre výpočtovo náročnejšie, ale presnejšie predikcie odporúčame použiť model *MA(30)*, ktorý je približne 5x výpočtovo náročnejší ako model *AR(1)*. Lepšie ale výpočtovo náročnejšie predikcie je možné dosiahnuť pomocou algoritmov vyberajúcich najlepší model zo všetkých dostupných modelov. V takom prípade sa dĺžka výpočtu predikcie rovná sume dĺžiek výpočtov jednotlivých predikčných algoritmov (tie je možné spustiť paralelne na viacerých procesoroch) a dĺžke výpočtu hodnotiacej funkcie použitej algoritmom pre výber najlepšieho modelu.

5.7 Predikčné intervaly

V dodatku F uvádzame grafy simulácie predikcie dátového toku spolu s predikčnými intervalmi. Grafy F.1 a F.2 obsahujú simulácie nad rovnakými dátami, z linky Praha Hradec Králové siete CESNET2. Prvý graf je predikovaný pomocou modelu *MA(25)*, zatiaľ čo druhý graf bol predikovaný pomocou algoritmu vyberajúceho najlepší model pomocou funkcie RMSE. Zatiaľ čo v prvom grafe predikčné intervaly takmer vôbec nezachytávajú sezónnosti, v druhom grafe je vidieť, že algoritmu sa úspešne darilo predikovať sezónnosti v dátach a zahrnúť to do predikčných intervalov. Každopádne predikčné intervaly v oboch prípadoch dobre zachytávajú rozsah budúcich predpovedaných hodnôt, bez ohľadu na použitý model.

Ešte väčší rozdiel v šírke predikčných intervalov v závislosti na výbere predikého modelu je možné pozorovať v grafoch F.3 a F.4 zo simulácie predikcie na dátach z linky AMS-IX Praha. Opäť platí, že aj pri výbere nevhodného modelu budú predikčné intervaly predikovať správny rozsah, v ktorom sa budú nachádzať budúce hodnoty.

Grafy F.5 a F.6 zobrazujú rovnakú dátovú linku a rovnaké predikčné modely ako predchádzajúce 2 grafy, predpoveď bola vykonávaná o 5 období dopredu. V predchádzajúcich 2 grafoch bola robená na 1 obdobie dopredu. Pri oboch predikčných modeloch sú predikčné intervaly pri predpovedi o 5 období dopredu znateľne širšie ako pri predpovedi o jedno obdobie dopredu. Predpoveď na dlhšie obdobie dopredu je totiž spojená s väčšou neurčitosťou. Preto sa rozšíril interval, v ktorom očakávame budúcu hodnotu.

V grafoch F.7 a F.8 uvádzame ešte dlhodobejšiu predpoveď, o 10 období dopredu. Opäť je predpoveď vykonávaná pomocou predikčných modelov *MA(25)* a algoritmu vyberajúcom najlepší model pomocou hodnotiacej funkcie RMSE. V grafoch je badateľný rozdiel v šírke a tvare predikčných intervalov. V prípade modelu *MA(25)* model opäť nezachycuje sezónnosť, zatiaľ čo algoritmus vyberajúci najlepší model vybral model s umelými premennými, ktorému sa podarilo detekovať sezónnosť a predikovať predikčné pomerne presne budúce hodnoty.

Predikčné intervaly splnili naše očakávania. Bez ohľadu na model predikujú hodnoty, v ktorých je možné očakávať budúce hodnoty (ak sa nezmení charakteristika dát). Pri nesprávne zvolenom modele sú širšie, čo vyjadruje zvýšenú neurčitosť predikcie. Taktiež zahrňujú zvýšenú neurčitosť pri predikovaní o viac období dopredu.

5.8 Detekcia anomálií

Experimenty simulujúce detekciu anomálií budú mať za cieľ analyzovať algoritmy detekujúce anomálie, zistiť ich silné a slabé stránky a určiť smer ďalšieho vývoja. Pri simulácii detekcie anomálií sme využívali len simulované sezónne dáta s pridanými anomáliami. Nekladieme si za cieľ vybrať najlepší algoritmus detekujúci anomálie. K takýmto záverom nemáme dostupné dáta s identifikovanými anomáliami. Simulácie na simulovaných dátach nepovažujeme za dostatočne vypovedajúce o povahe algoritmov v reálnom nasadení.

Simulované anomálie pozostávali v náhlom zvýšení, prípadne znížení dátových tokov. Tým sme simulovali výrazné zvýšenie dátovej prevádzky, prípadne výpadok dátovej linky. Pri simuláciách sme nastavili parameter citlivosti algoritmov detekujúcich anomálie na hodnoty 0; 0,2; 0,4; 0,6; 0,8 a 1 (zoradené od najnižšej po najvyššiu). V simulácii boli použité nasledujúce algoritmy:

- Jednoduchý jednostavový algoritmus detekujúci na základe 3σ, označený ako *standardDeviationAnomalyDetector*
- Dvojstavový algoritmus detekujúci na základe 3σ, v grafe *twoStepStandardDeviationAnomalyDetector*
- Dvojstavový algoritmus využívajúci predikčné intervaly a model MA(5), označený ako *ma5.predictor_pi_detector*
- Dvojstavový algoritmus využívajúci predikčné intervaly a model MA(15) s menom *ma15.predictor_pi_detector*
- Dvojstavový algoritmus využívajúci predikčné intervaly a model MA(25), označený ako *ma25.predictor_pi_detector*
- Dvojstavový algoritmus využívajúci predikčné intervaly a automaticky identifikovaný model AR pomocou funkcie pacf, v grafe uvádzaný pod menom *experimental1.predictor_pi_detector*
- Dvojstavový algoritmus využívajúci minimálnu a maximálnu hodnotu predchádzajúcich pozorovaní, označený ako *pastQuantilePIEstimator_pi_detector*
- Dvojstavový algoritmus s predikčnými intervalmi generovanými algoritmom pre výber najlepšieho modelu na základe hodnotiacej funkcie RMSE, označený ako *predictorSelectorPIEstimator_pi_detector*

Graf G.1 znázorňuje simuláciu detekovania anomálie pomocou algoritmu vyberajúceho zo všetkých dostupných modelov pri citlivosti 0,6. Približne takto by mal vyzerať priebeh správne detekovanej anomálie. Problém je ale vo voľbe parametru citlivosti, detailnejšie prebranej v kapitole 4.3.6. Na grafe G.2 je simulácia toho istého algoritmu na rovnakých dátach s parametrom citlivosti 1 (najvyššia citlivosť). Tu algoritmus v prvej tretine nesprávne označil normálnu hodnotu ako anomáliu. Následne nezačlenil túto hodnotu do hodnôt používaných pre výpočet predikčného intervalu. V čase 1970-5-11 opäť dochádza k nesprávnemu detekovaniu anomálie. Opäť nie je zahrnutá táto nesprávne detekovaná anomália do hodnôt pre výpočet predikčného intervalu a tak algoritmus stále detekuje nesprávne anomálie. Ak by užívateľ označil prvú anomáliu ako nesprávne detekovanú, bola by pri výpočte zahrnutá do predikčných intervalov a algoritmus by sa správal pri ďalších detekciach ako algoritmus na grafe 4.3.6.

5.8.1 ROC krivky detektorov anomáií

Pre analýzu algoritmov detekujúcich anomálie používame v našej práci ROC krivky. Vzhľadom k rozsahu nebudeme v tejto práci definovať a popisovať ROC krivky, dobrý popis sa nachádza napríklad v publikácii [10]. Skrátený popis sa nachádza na začiatku prílohy G.1.

Výsledky algoritmov detekujúcich anomálie na simulovaných dátach s jednou dlho trvajúcou anomáliou uvádzame v grafe G.4. Nasleduje vyhodnotenie jednotlivých algoritmov:

- Jednoduchý algoritmus *pastQuantilePIEstimator_pi_detector* s parametrom citlivosti 0,8 dosahoval najlepšie výsledky. Vyššia citlivosť neviedla k zlepšeniu.
- Algoritmus *twoStepStandardDeviationAnomalyDetector* dosahoval dobré výsledky pri parametre citlivosti 1.
- Algoritmus *predictorSelectorPIEstimator_pi_detector* pri parametre citlivosti 0,6 dosahoval tiež dobré výsledky.
- Ďalším algoritmom sa tiež podarilo správne detekovať túto anomáliu, v polovičnom čase trvania anomálie ale začali skutočnú anomáliu detekovať ako normálny stav.

Ďalšie výsledky algoritmov detekujúcich anomálie na simulovaných dátach s rozdielnymi parametrami ako v predchádzúcom prípade uvádzame v grafe G.5. Opäť nasleduje vyhodnotenie jednotlivých algoritmov:

- Algoritmy *twoStepStandardDeviationAnomalyDetector* pri citlivosti 0,8 a *pastQuantilePIEstimator_pi_detector* pri citlivosti 0,6 dosahovali najlepšie výsledky. S vyššou citlivosťou dochádzalo k väčšiemu počtu chybne detekovaných anomálií.
- Dobré výsledky dosahoval tiež *predictorSelectorPIEstimator_pi_detector* pri citlivosti 0,6. Nedokázal ale detekovať anomáliu počas celého trvania anomálie.

ROC krivka má štandardne rastúci trend. So zvyšujúcou sa senzitivitou stúpa aj počet nesprávne detekovaných anomálií. V grafe G.6 sme narazili pri algoritme *predictorSelectorPIEstimator_pi_detector* na opačný jav, krivka nemala len rastúci trend. To bolo spôsobené nedetekovanou anomáliou. Algoritmus s nastavením citlivosti 0,6 nedetekoval anomáliu. Následne bola táto anomália zahrnutá do dát pre výpočet lineárneho modelu. Predikčné intervaly sa rozšírili a algoritmus nedetekoval skutočné anomálie. Pri citlivosti 1 tento algoritmus detekoval všetky anomálie. Predikčné intervaly zodpovedali budúcim normálnym hodnotám a algoritmus nedetekoval žiaden normálny stav ako anomáliu. Z tohto príkladu je vidieť, že nastavenie citlivosti algoritmov vedie k rozdielnym kvalitatívnym výsledkom, pričom citlivosť sa nemusí správať lineárne. Tento problém by vyriešilo ručné potvrdzovanie, či na sieti naozaj došlo k anomálii, alebo nie. V tomto prípade dosahoval najlepšie výsledky už spomínaný algoritmus *predictorSelectorPIEstimator_pi_detector* pri parametre citlivosti 1. Druhým najlepším bol *twoStepStandardDeviationAnomalyDetector* s citlivosťou 0,8.

Pre praktické nasadenie odporúčame kombináciu algoritmov *twoStepStandardDeviationAnomalyDetector* s citlivosťou 0,8, *pastQuantilePIEstimator_pi_detector* s citlivosťou okolo 0,8 a *predictorSelectorPIEstimator_pi_detector* s citlivosťou medzi $0,6 - 0,8$ v závislosti na požadovanej celkovej citlivosti. Každý algoritmus pracuje na inom princípe a na rôznych typoch dát dosahujú rôzne výsledky. Pre stacionárne dáta bez trendu a sezónnosti funguje dobre algoritmus *twoStepStandardDeviationAnomalyDetector*. Pre dáta s trendom, prípadne sezónnosťou

je vhodné použiť model *predictorSelectorPIEstimator_pi_detector*. Vďaka možnosti kombinovať detektory anomálií popísanej v kapitole 4.3.5 je možné použiť väčšie množstvo detektorov.

Tieto experimenty je nutné brať s určitou rezervou. Boli vykonávané na simulovaných dátach, pričom anomálie boli náhodne generované. Pre praktické použitie by bolo vhodné vykonať experimenty ešte raz na reálnej dátovej sade s označenými skutočnými anomáliami. Tieto experimenty sme neboli schopní simulovať, pretože sme neboli schopní získať časové rady dátových tokov s označenými anomáliami.

5.9 Zhrnutie výsledkov

Pre vytváranie bodovej predikcie odporúčame algoritmus vyberajúci najlepší model na základe hodnotiacej funkcie (*RMSE selector*), prípadne model váženého priemeru všetkých dostupných modelov (*All models average*). Ostatné nami implementované modely boli schopné vytvárať kvalitné predpovede len na určitom type dát.

Pre intervalové predikcie odporúčame empirické predikčné intervaly v kombinácii s algoritmom vyberajúcim model podľa hodnotiacej funkcie (*RMSE selector*).

Algoritmy detekujúce anomálie by bolo vhodné pred nasadením experimentálne vyhodnotiť na reálnych dátach s označenými anomáliami. Do praxe odporúčame nasadenie algoritmov *twoStepStandardDeviationAnomalyDetector* a ďalej algoritmy *pastQuantilePIEstimator_pi_detector* a *predictorSelectorPIEstimator_pi_detector*.

5.10 Budúci vývoj

5.10.1 Parametrizácia algoritmu detekujúceho anomálie pridávaním stavov

Nami navrhnutý dvojstavový algoritmus detekujúci anomálie popísaný v kapitole 4.3.2 používal k detekovaniu anomálie 2 stavy, možná anomália a detekovaná anomália. Do stavu možná anomália algoritmus prešiel, ak aktuálna hodnota dátového toku prekročila predikčné intervaly. Do stavu detekovaná anomália prešiel ak bol v stave možná anomália a ďalšia hodnota opäť prekročila predikčné intervaly. Algoritmus detekoval anomáliu najskôr po 2 časových obdobiach.

V niektorých prípadoch nechceme náhle zvýšenie dátového toku po dobu dvoch časových období označovať ako anomáliu. Ak zberáme údaje o dátových tokoch s periódou 1 záznam za 5 sekúnd, 2 obdobia odpovedajúce 10 sekundám sú pre nás príliš krátky okamih na to, aby sme posúdili, či sa jedná o anomáliu a či treba nejako zakročiť.

Navrhujeme preto do budúcna rozšíriť dvojstavový algoritmus o možnosť konfigurácie počtu stavov, po ktorých prejde algoritmus zo stavu možná anomália do stavu detekovaná anomália. Algoritmus by fungoval nasledovne: ak dátový tok prekročí predikčné intervaly, algoritmus prejde do stavu možná anomália. Ak následne po dobu N období (N je nastaviteľný parameter počtu období pre detekciu anomálií) bude hodnota mimo predikčný interval, algoritmus prejde do stavu detekovaná anomália. Ak sa hodnota dátového toku vráti počas týchto N období do predikčných intervalov, algoritmus prejde do normálneho stavu a nebude detekovať anomáliu.

Toto rozšírenie navrhujeme použiť len v prípade, že dátový tok prekročí maximum predikčného intervalu – čiže dátový tok bude vyšší ako predpoveď. V prípade dátového toku nižšieho ako predpoveď odporúčame okamžite detekovať anomáliu. V takom prípade sa pravdepodobne jedná o výpadok siete a obsluha by mala byť okamžite informovaná o tejto

anomálii.

5.10.2 Sledovanie portov a detekovanie anomálií na portoch

Algoritmy pre detekovanie anomálií by bolo možné použiť na sledovanie dátových tokov na konkrétnych portoch dátových liniek siete. Ak sledujeme celkový dátový tok linky, pravdepodobne nebudeme schopní detekovať výpadok jednej služby (napr. SMTP, HTTP v prípade HTTP servera). Ak sa zameriame na dátový tok na jednotlivých portoch, budeme schopní detekovať výpadky jednotlivých služieb, prípadne výrazný nárast aktivity na týchto portoch. Tento prístup môže byť obmedzený metódou, ktorou moniturujeme sieť. Ak získavame dáta pomocou protokolu SNMP zo smerovačov v sieti, dostávame agregované veličiny a nie sme schopní získať podrobnosti o jednotlivých portoch.

Algoritmy uvedené v tejto práci nebude nutné nijakým spôsobom modifikovať. Zmena bude musieť nastať v programe zberajúcom dáta v sieti.

5.10.3 Sledovanie vnútornej topológie siete

V prípade znalosti topológie siete je možné upraviť modely a pridať do nich túto informáciu. Pre tento účel by bolo možné použiť model panelových dát. Výhodou takéhoto modelu by mohli byť lepšie predikčné schopnosti. Ak by došlo k zvýšenej záťaži na jednej konkrétnej linke, model by mohol byť schopný predpovedať celkovú zvýšenú záťaž siete. Ďalej by mohol model predikovať pri výpadku linky preťaženie inej linky (pretože pakety budú presmerované cez túto linku).

Modely obsahujúce topológiu siete by ďalej bolo možné použiť aj pre simulácie vyťaženia siete, prípadne simulácie výpadkov konkrétnych liniek. Pravdepodobne by sme narazili na problém s použitím lineárnych modelov. Dátové toky v sieťach sa kvôli hornému a spodnému obmedzeniu dátových liniek (nulový dátový tok a maximálna prenosová rýchlosť siete) nebudú dať modelovať pomocou lineárneho modelu.

5.10.4 Pridávanie ďalších informácií do siete

Naše modely by bolo tiež možné rozšíriť o znalosti konkrétnych špecifických znalostí danej linky. Napríklad v prípade wifi smerovača by mohol byť pridaný do modelu počet klientov pripojených k tomuto smerovaču. Očakávame, že s vyšším počtom klientov na wifi sieti bude tiež stúpať prevádzka na linke spájajúcej wifi smerovač so zvyškom siete. Ďalšou veličinou, ktorá by mohla byť zaradená do modelov je ping, počet zahodených paketov, prípadne vyťaženie procesora smerovača. Do modelov by mohla byť pridaná aj fyzická charakteristika dátovej linky (o aké prenosové médium sa jedná) a maximálna kapacita linky.

Kapitola 6

Záver

V tejto práci sme sa zaoberali bodovou a intervalovou predikciou dátových tokov sietí a následným detekovaním anomálií v sieťach na základe dátových tokov. Našou prioritou bolo vytvorenie algoritmu, ktorý bude vytvárať predikcie bez predchádzajúcej konfigurácie a zároveň bude fungovať na dátových linkách s rozdielnou charakteristikou toku dát. Od algoritmu sme ďalej požadovali, aby bol dostatočne rýchly.

Pred začatím navrhovania predikčných algoritmov sme spravili v kapitole 3 analýzu časových radov dátových tokov v sieti CESNET2. V tejto analýze sme sa zamerali na hľadanie vzorov v dátach. V niektorých časových radoch dátových tokov sme našli napríklad týždňové sezónnosti, prípadne denné sezónnosti. Tieto poznatky nám neskôr poslúžili pri implementácii predikčných algoritmov. Analýza ďalej ukázala, že nami skúmané časové rady majú rozdielne štatistické charakteristiky. Približne polovica nami skúmaných časových radov bola stacionárna, pri druhej polovici sme zamietli túto hypotézu. Z tejto analýzy vyplynulo, že časové rady dátových tokov môžu mať rozdielne charakteristiky a s vysokou pravdepodobnosťou nebude možné použiť jeden štatistický model k predpovedaniu všetkých typov časových radov.

K predikcii dátových tokov sme v kapitole 4.1 navrhli niekoľko štatistických modelov. Konkrétne sa jednalo o autoregresný model, autoregresný model s umelými premennými a model kĺzavého priemeru. Ďalej sme navrhli niekoľko algoritmov, ktoré automaticky nastavovali parametre modelov, prípadne vyberali vhodné členy modelov. Pre experimentálne vyhodnotenie predikčných algoritmov sme simulovali predikcie na dátovej sade zo siete CESNET2 v kapitole 5. Z výsledkov vyplynulo, že najlepšie výsledky pri predikovaní dosahovali algoritmy automaticky vyberajúce model na základe predchádzajúcich výsledkov predikcií, prípadne algoritmy kombinujúce viacero predikčných modelov. V tejto kapitole sa ďalej nachádza porovnanie výpočtovej náročnosti týchto algoritmov.

Kapitola 4.2 sa zaoberala intervalovými predikciami. Vzhľadom k našim požiadavkám na nulovú konfiguráciu algoritmov a schopnosť predpovedať na rôznych dátových tokoch sme nemohli použiť intervalové predikcie založené na určitých predpokladoch o rozložení predikčných chýb. V tejto kapitole sme tiež analyzovali predikčné chyby z predchádzajúcich experimentov bodových predpovedí. Potvrdili sa naše predpoklady, predikčné chyby mali rozdielne rozloženia. Preto sme použili pre vytváranie predikčných chýb empirické predikčné intervaly. Tie sa nezakladajú na predpokladoch o rozložení predikčných chýb. Predikčné intervaly sú počítané z historických predikčných chýb daného modelu na dátovej linke, pre ktorú vytvárame predikciu. Výhodou tejto techniky je väčšia flexibilita. Predikčné intervaly boli dostatočne široké aj v prípade nesprávne vybratého modelu pre bodovú predikciu. Nevýhodou je vyššia pamäťová náročnosť, algoritmus si musí držať v pamäti predchádzajúce predikčné chyby. Ďalej je nutné, aby algoritmus pred začatím konštruovania prvých intervalových predikcií vytvoril niekoľko bodových predikcií (aspoň 30), z ktorých následne spočíta predikčné chyby.

Intervalovú predikciu sme následne použili v kapitole 4.3 na detekovanie anomálií v dátových sieťach. Za anomáliu sme považovali náhle zvýšenie, prípadne zníženie dátových tokov. Nami navrhnutý algoritmus vytváral predikčné intervaly a ak aktuálna hodnota dátového toku prekročila predikčné intervaly, označoval algoritmus túto hodnotu ako anomáliu. Algoritmus sme porovnávali s jednoduchou technikou detekovania anomálií na časových radoch. Táto technika spočívala vo vytvorení obdoby predikčného intervalu na základe násobenia smerodajnej odchýlky. Došli sme k záveru, že pri niektorých typoch anomálií je lepšie použiť nami navrhnutý algoritmus, v niektorých prípadoch fungovala lepšie primitívna technika. Algoritmy pre detekciu anomálií je možné kombinovať a použiť obe techniky naraz.

V poslednej časti 5.10 navrhujeme ďalšie použitie nami navrhnutých algoritmov. Pomocou algoritmov by šli predikovať a detekovať anomálie na metrikách ako napríklad počeť užívateľov siete, využitie zdieľaných zdrojov, prípadne vyťaženie výpočtových zdrojov. Ďalej navrhujeme použitie nami navrhnutých algoritmov na detekovanie anomálií dátových tokov na konkrétnych portoch dátových liniek. V budúcnosti by bolo tiež možné implementovať modely, ktoré by do výpočtov zahrňovali viac vstupných informácií, ako napríklad počet užívateľov siete, prípadne topológiu siete. Znalosť ďalších informácií by totiž mohla viesť ku kvalitnejším predpovediam.

Literatúra

[1] Ahmed, T.; Coates, M.; Lakhina, A.: Multivariate online anomaly detection using kernel recursive least squares. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, IEEE, 2007, s. 625–633.

[2] Ahmed, T.; Oreshkin, B.; Coates, M.: Machine learning approaches to network anomaly detection. In *Proceedings of the 2nd USENIX workshop on Tackling computer systems problems with machine learning techniques*, SYSML'07, Berkeley, CA, USA: USENIX Association, 2007, s. 7:1–7:6.

[3] Armstrong, J.: *Principles of Forecasting: A Handbook for Researchers and Practitioners*. International Series in Operations Research & Management Science, Springer, 2001, ISBN 9780792379300.

[4] Armstrong, J. S.; Collopy, F.: Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, ročník 8, č. 1, 1992: s. 69–80.

[5] Beigi, M.; Verma, D.: Network prediction in a policy-based IP network. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, ročník 4, IEEE, 2001, s. 2522–2526.

[6] Breunig, M.; Kriegel, H.-P.; Ng, R. T.; aj.: LOF: Identifying Density-Based Local Outliers. In *PROCEEDINGS OF THE 2000 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA*, ACM, 2000, s. 93–104.

[7] Brockwell, P.; Davis, R.: *Time Series: Theory and Methods*. Springer Series in Statistics, Springer, 2009, ISBN 9781441903198.

[8] Brooks, C.: *Introductory Econometrics for Finance*. Cambridge University Press, 2008.

[9] Brutlag, J. D.: Aberrant Behavior Detection in Time Series for Network Monitoring. In *Proceedings of the 14th USENIX conference on System administration*, LISA '00, Berkeley, CA, USA: USENIX Association, 2000, s. 139–146.

[10] Bushberg, J.; Seibert, J.; Leidholdt, E.; aj.: *The Essential Physics of Medical Imaging*. Wolters Kluwer Health, 2011, ISBN 9781451153941.

[11] Chatfield, C.: Calculating Interval Forecasts. *Journal of Business and Economic Statistics*, ročník 11, č. 2, 1993: s. pp. 121–135, ISSN 07350015.

[12] Chatfield, C.: *Time-series forecasting*. Chapman and Hall/CRC, 2002.

[13] Deri, L.; Suin, S.; Maselli, G.: Design and implementation of an anomaly detection system: An empirical approach. In *Proceedings of Terena TNC*, ročník 2003, Citeseer, 2003, str. 2001.

[14] Feng, H.; Shu, Y.: Study on network traffic prediction techniques. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, ročník 2, IEEE, 2005, s. 1041–1044.

[15] FRED, F. R. E. D.: Federal Reserve Bank of St. *Louis: Exchange Rates*, ročník 5, 2010.

[16] Ghaderi, M.: On the relevance of self-similarity in network traffic prediction. 2003.

[17] Hanbanchong, A.; Piromsopa, K.: SARIMA based network bandwidth anomaly detection. In *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, IEEE, 2012, s. 104–108.

[18] Heij, C.; de Boer, P.; Franses, P. H.; aj.: *Econometric Methods with Applications in Business and Economics*. Oxford University Press, 2004.

[19] Hinich, M. J.; Molyneux, R. E.: Predicting information flows in network traffic. *Journal of the American Society for Information Science and Technology*, ročník 54, č. 2, 2003: s. 161–168.

[20] Hu, L.; Che, X.: Design and implementation of bandwidth prediction based on grid service. In *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, IEEE, 2008, s. 45–52.

[21] Huang, L.; Nguyen, X.; Garofalakis, M.; aj.: In-network PCA and anomaly detection. *Advances in Neural Information Processing Systems*, ročník 19, 2007: str. 617.

[22] Hussain, F.; Kalim, U.; Latif, N.; aj.: Using End-to-End Bandwidth Estimates for Anomaly Detection beyond Enterprise Boundaries. 2010.

[23] Janert, P. K.: *Gnuplot in Action: Understanding Data with Graphs*. Manning Publications Co., 2009.

[24] Kim, H.-J.; Tomppo, E.: Model-based prediction error uncertainty estimation for k-nn method. *Remote Sensing of Environment*, ročník 104, č. 3, 2006: s. 257–263, ISSN 0034-4257.

[25] Kim, S. S.; Reddy, A. N.; Vannucci, M.: Detecting traffic anomalies using discrete wavelet transform. In *Information Networking. Networking Technologies for Broadband and Mobile Networks*, Springer, 2004, s. 951–961.

[26] Lazarevic, A.; Ertoz, L.; Kumar, V.; aj.: A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the third SIAM international conference on data mining*, ročník 3, Siam, 2003, s. 25–36.

[27] Luo, Y.; Ansari, N.: Limited sharing with traffic prediction for dynamic bandwidth allocation and QoS provisioning over Ethernet passive optical networks. *J. Opt. Netw.*, ročník 4, č. 9, Sep 2005: s. 561–572, doi:10.1364/JON.4.000561.

[28] Madden, G.; Tan, J.: Forecasting international bandwidth capacity using linear and ANN methods. Taylor & Francis, 2008, s. 1775–1787.

[29] McGarry, M. P.; Haddad, R.; McAlarney, J.: A new approach to video bandwidth prediction. In *Ultra Modern Telecommunications & Workshops, 2009. ICUMT'09. International Conference on*, IEEE, 2009, s. 1–5.

[30] Mertz, D.: Statistical programming with R, Part 3: Reusable and object-oriented programming. IBM, January 2006.

[31] Nychis, G.; Sekar, V.; Andersen, D. G.; aj.: An empirical evaluation of entropy-based traffic anomaly detection. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, ACM, 2008, s. 151–156.

[32] Rehák, M.; Pěchouček, M.; Bartoš, K.; aj.: CAMNEP: An intrusion detection system for high-speed networks. *Progress in Informatics*, 2008.
URL `http://www.nii.ac.jp/pi/`

[33] Roesch, M.: Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX conference on System administration*, LISA '99, Berkeley, CA, USA: USENIX Association, 1999, s. 229–238.

[34] Rutka, G.: Neural network models for Internet traffic prediction. *Proceedings of Electronics and Electrical Engineering, Lithuania*, ročník 4, č. 68, 2006: s. 55–58.

[35] Rutka, G.; Lauks, G.: Study on internet traffic prediction models. *Proceedings of Electronics and Electrical Engineering, Lithuania, Kaunas*, 2007: s. 21–23.

[36] SAID, S. E.; DICKEY, D. A.: Testing for Unit Roots in Autoregressive-Moving Average Models of Unknown Order. *Biometrika*, ročník 71, č. 3, 1984: s. 599–607, doi:10.1093/biomet/71.3.599.

[37] Seber, G.; Lee, A.: *Linear Regression Analysis*. Wiley Series in Probability and Statistics, Wiley, 2012, ISBN 9781118274422.

[38] Sevcik, P.: Internet bandwidth: its time for accountability. *Business Communications Review*, ročník 31, č. 1, 2001: str. 10.

[39] Shadbolt, J.; Taylor, J.: *Neural Networks and the Financial Markets: Predicting, Combining, and Portfolio Optimisation*. Perspectives in Neural Computing Series, Springer London, Limited, 2002, ISBN 9781852335311.

[40] Slavicek, K.; Novak, V.; Ledvinka, J.: CESNET Fiber Optics Transport Network. In *Networks, 2009. ICN '09. Eighth International Conference on*, 2009, s. 403–408, doi:10.1109/ICN.2009.80.

[41] Song, S.; Keleher, P.; Bhattacharjee, B.; aj.: Brief announcement: Decentralized network bandwidth prediction. In *Distributed Computing*, Springer, 2010, s. 198–200.

[42] Song, S.; Keleher, P.; Bhattacharjee, B.; aj.: Decentralized, accurate, and low-cost network bandwidth prediction. In *INFOCOM, 2011 Proceedings IEEE*, IEEE, 2011, s. 6–10.

[43] Swanson, D.; Tayman, J.; Bryan, T.: MAPE-R: a rescaled measure of accuracy for cross-sectional subnational population forecasts. *Journal of Population Research*, ročník 28, č. 2-3, 2011: s. 225–243, ISSN 1443-2447, doi:10.1007/s12546-011-9054-5.

[44] Thottan, M.; Ji, C.: Anomaly detection in IP networks. *Signal Processing, IEEE Transactions on*, ročník 51, č. 8, 2003: s. 2191–2204.

[45] Venables, W. N.; Smith, D. M.; Team, R. D. C.: An introduction to R. 2002.

[46] Vera, G.; Jansen, R.; Suppi, R.: R/parallel - speeding up bioinformatics analysis with R. *BMC Bioinformatics*, ročník 9, č. 1, 2008: str. 390, ISSN 1471-2105, doi:10.1186/1471-2105-9-390.

[47] Verbeek, M.: *A guide to modern econometrics*, ročník 2. John Wiley & Sons New York, 2004.

[48] Wang, H.; Zhang, D.; Shin, K. G.: Detecting SYN flooding attacks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, ročník 3, IEEE, 2002, s. 1530–1539.

[49] Wei, W. W.-S.: *Time series analysis*. Addison-Wesley Redwood City, California, 1994.

[50] Wolski, R.; Obertelli, G.; Allen, M.; aj.: Predicting Grid Resource Performance Online. In *Handbook of Nature-Inspired and Innovative Computing*, editace A. Zomaya, Springer US, 2006, ISBN 978-0-387-40532-2, s. 575–611.

[51] Wolski, R.; Spring, N. T.; Hayes, J.: The network weather service: a distributed resource performance forecasting service for metacomputing. Elsevier, 1999, s. 757–768.

[52] Xiang, L.; Ge, X.-H.; Liu, C.; aj.: A new hybrid network traffic prediction method. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, IEEE, 2010, s. 1–5.

Dodatok A

Executive summary

The goal of this work was to develop an algorithm for predicting data bandwidth in computer networks. The prediction algorithm requirements were specified by SolarWinds, the industrial partner of Faculty of Informatics. The prediction algorithm was supposed to be running without complicated configuration on all types and characteristics of computer networks. The algorithm was also required to have a good performance. Input data for this algorithm were historical time series of network bandwidth.

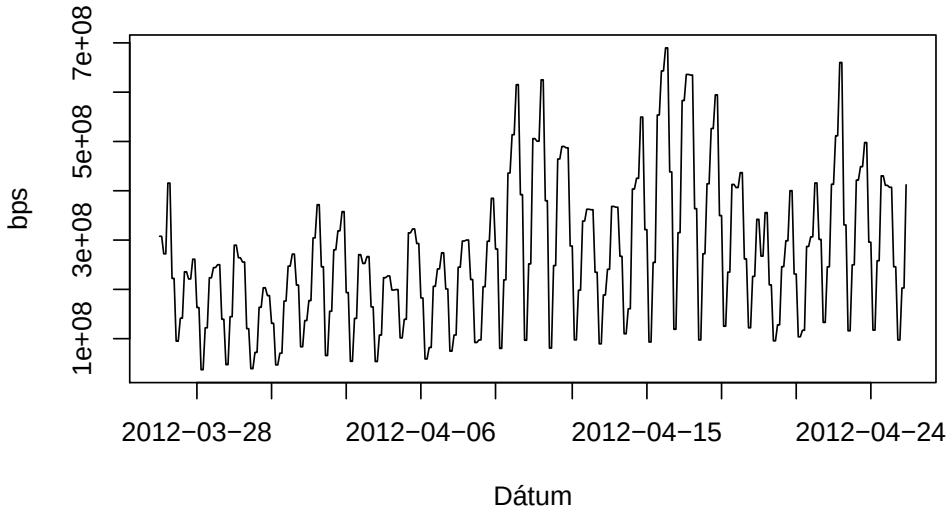
We have developed several statistical models, such as autoregressive model, autoregressive model with artificial variables and moving average model. We simulated prediction with these models on historical data from academic network CESNET2. Some of these models had good prediction performance on some networks, but none of them had good prediction performance on all types of network bandwidth data. Therefore we developed algorithm, which was able to select best model from given portfolio of models. We also developed algorithm, which was able to combine more models into one model with weighted average. Those algorithms had good prediction performance in our simulations. We recommend using them for automatic prediction of network bandwidth.

Next part of our work dealt with interval prediction. We analyzed prediction errors from our prediction models and find out, that standard formulas for computing prediction intervals were not applicable. Without previous knowledge of network bandwidth characteristics it is impossible to construct formula for computing prediction intervals. We proposed to use empirical prediction intervals based on previous prediction errors. This method had good results in our simulations.

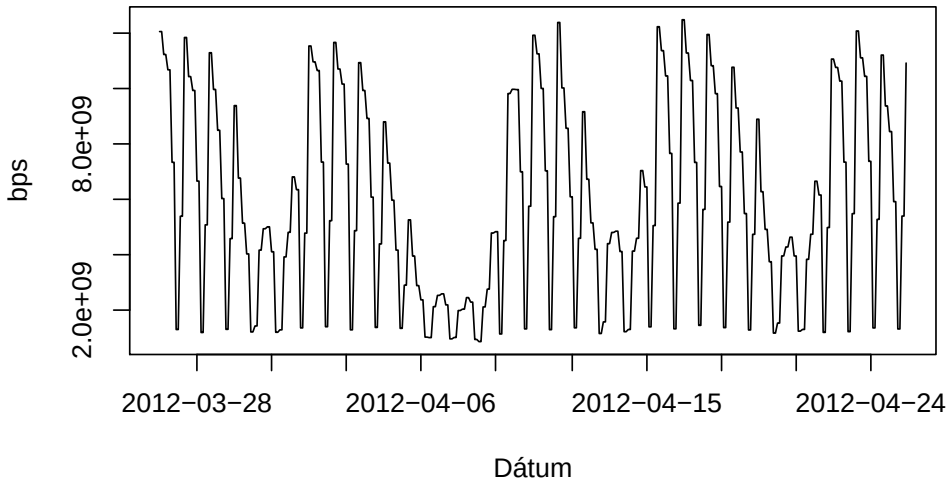
In the last part we used prediction intervals for detecting anomalies in network bandwidth. We compared this method with other primitive methods for anomaly detection. Results were ambiguous, quality of results depended on data and anomaly characteristics. Our recommendation is to use more methods for anomaly detection and combine their results.

Dodatok B

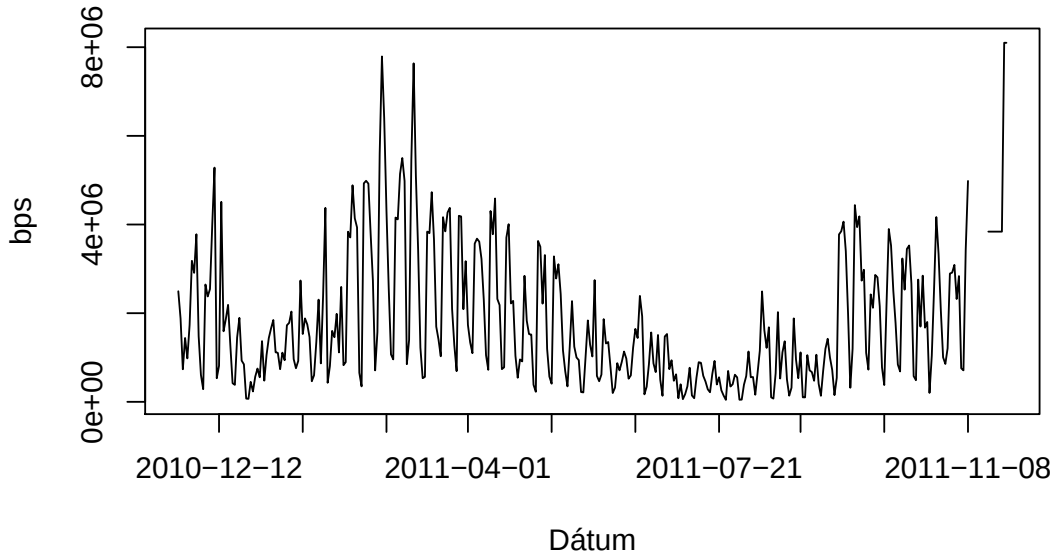
Vybrané časové rady dátových tokov siete CESNET2



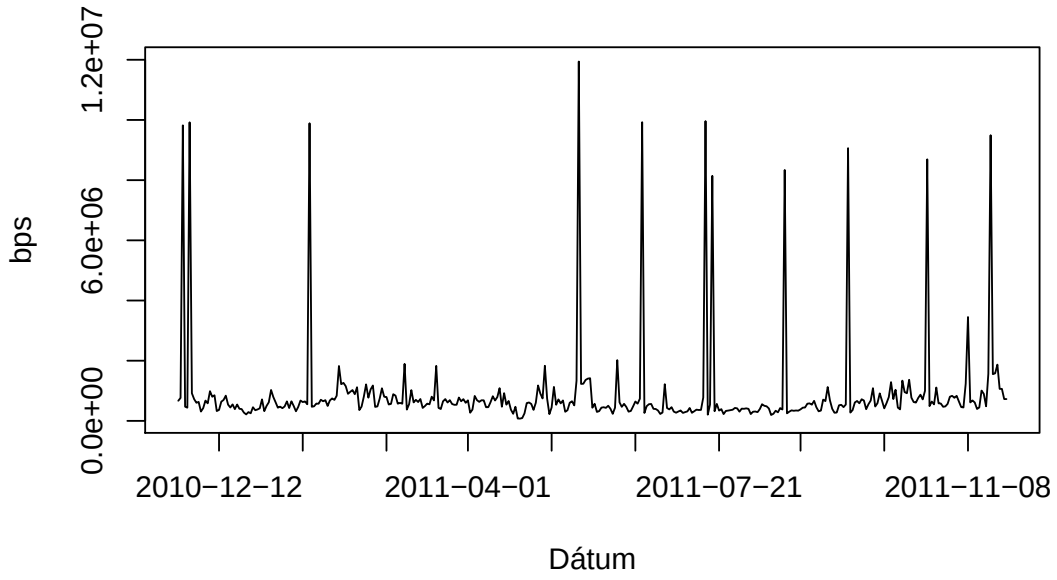
Obr. B.1: Dátový tok na sieti CESNET2 medzi Hradcem Králové a Prahou, granularita dát 1 záznam za hodinu



Obr. B.2: Dátový tok na sieti CESNET2 medzi NIXom a Prahou, granularita dát 1 záznam za hodinu



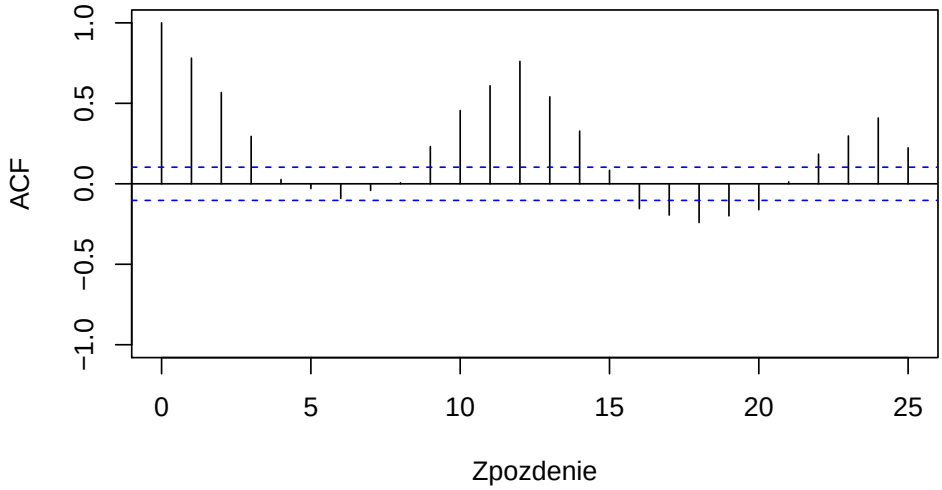
Obr. B.3: Dátový tok na sieti CESNET2 medzi Hradcem Králové a Libercom, granularita dát 1 záznam za deň



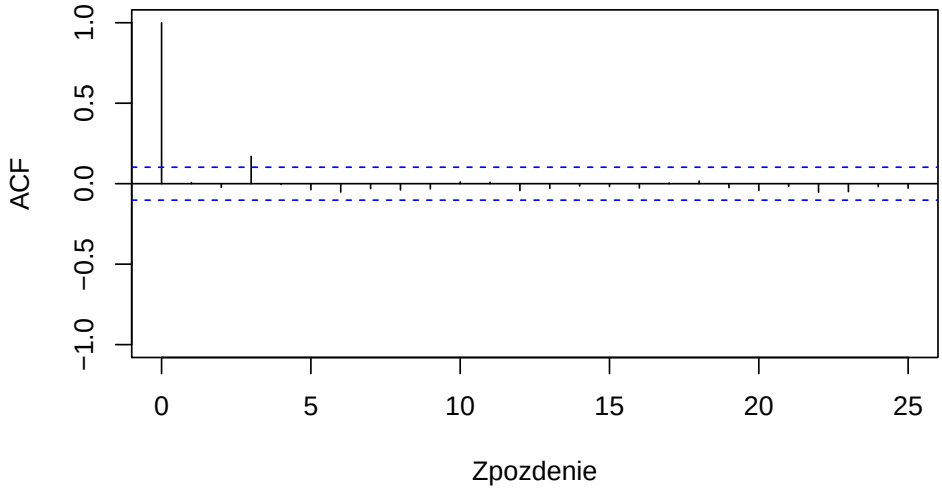
Obr. B.4: Dátový tok na sieti CESNET2 medzi Mostem a Ústí nad Labe, granularita dát 1 záznam za hodinu

Dodatok C

Autokorelačná funkcia vybraných dátových tokov siete CESNET2



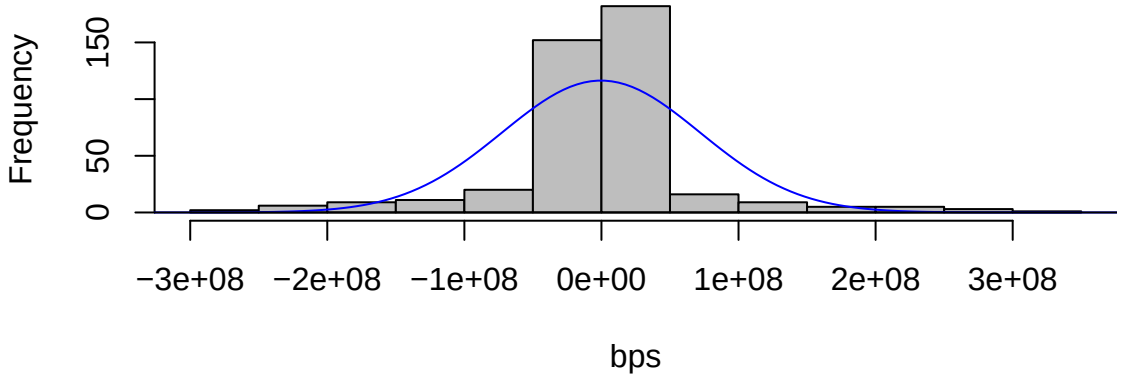
Obr. C.1: Autokorelačná funkcia dátového toku na sieti CESNET2 medzi Prahou a NIX, granularita dát 1 záznam za hodinu



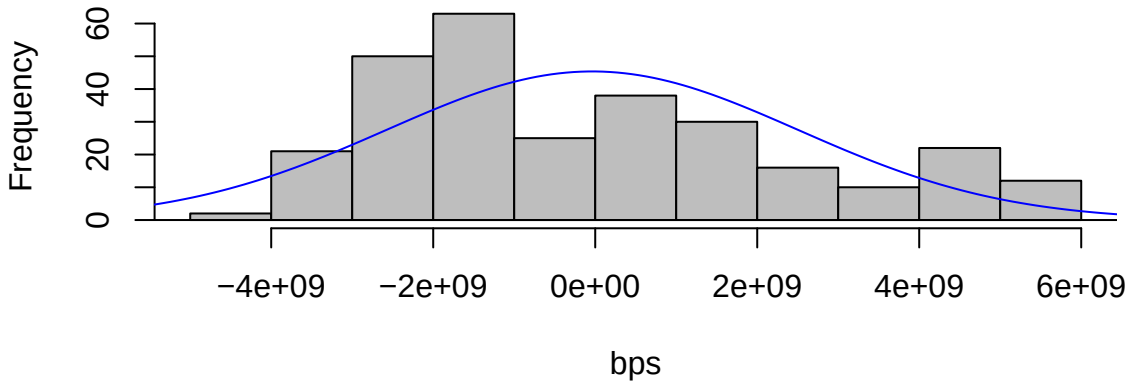
Obr. C.2: Autokorelačná funkcia dátového toku na sieti CESNET2 medzi Mostem a Ústí nad Labe, granularita dát 1 záznam za hodinu

Dodatok D

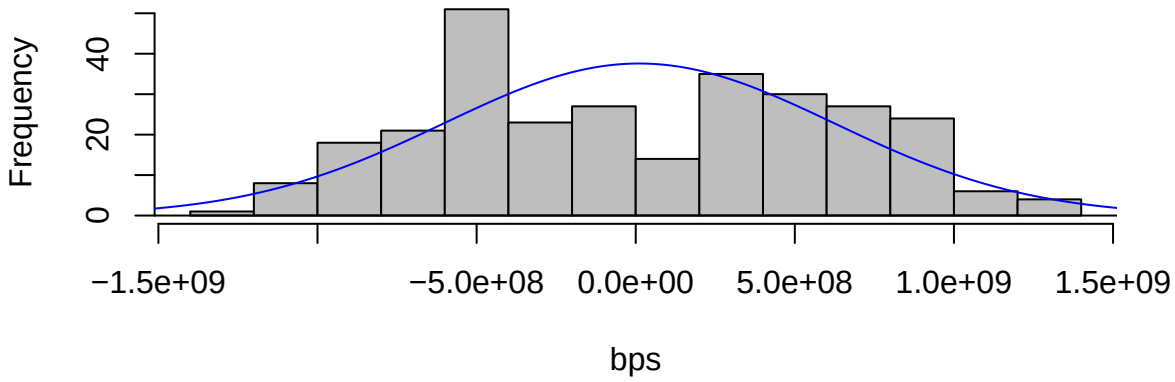
Histogramy predikčných chýb na vybraných dátových sadách



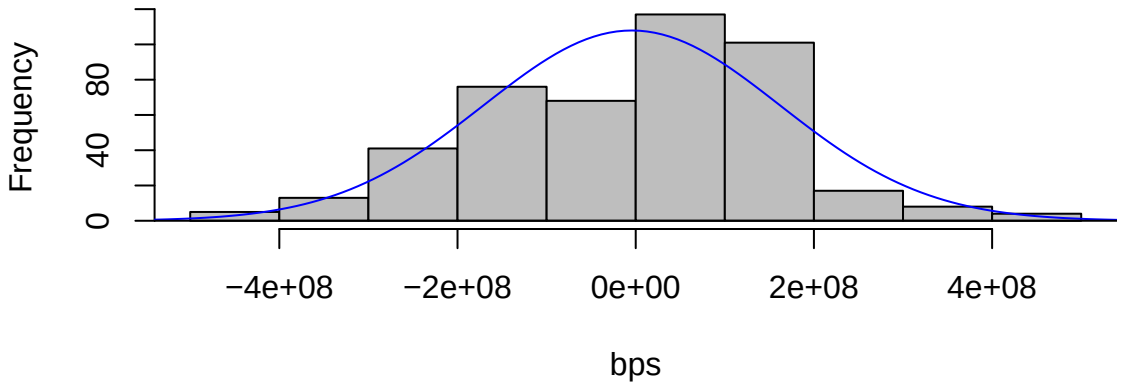
Obr. D.1: Histogram predikčných chýb modelu AR(1) na linke AMS-IX Praha s granularitou 1 záznam za 20 minút, predpovede na 1 obdobie dopredu, pri predpovediach bolo braných do úvahy 80 historických pozorovaní



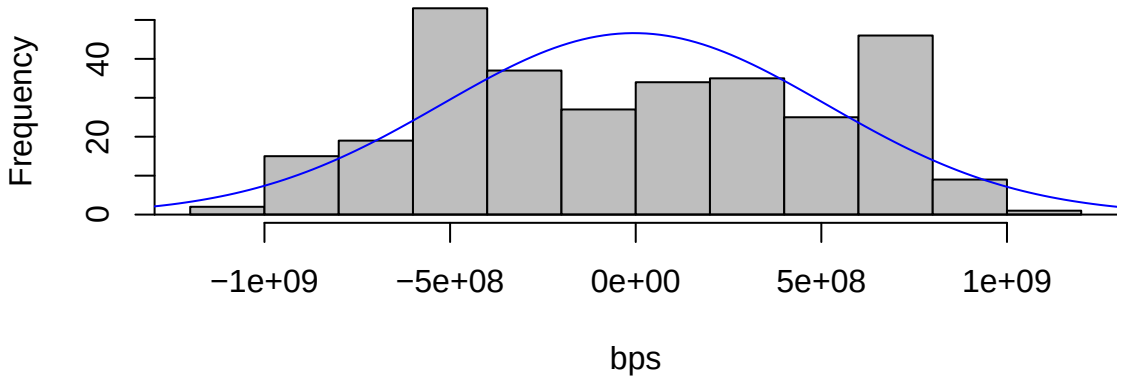
Obr. D.2: Histogram predikčných chýb modelu AR(1) na linke NIX Praha s granularitou 1 záznam za 12 hodín, predpovede na 1 obdobie dopredu, pri predpovediach bolo braných do úvahy 80 historických pozorovaní



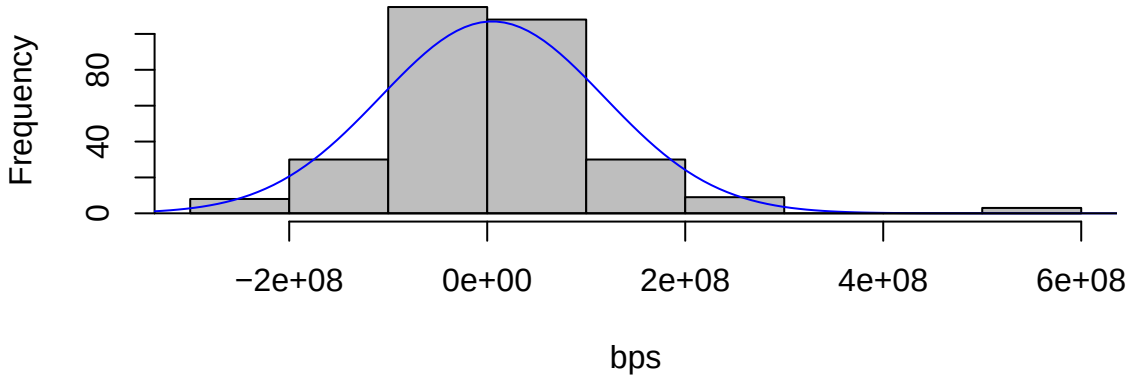
Obr. D.3: Histogram predikčných chýb modelu AR(1) na linke Praha NIX s granularitou 1 záznam za 12 hodín, predpovede na 1 obdobie dopredu, pri predpovediach bolo braných do úvahy 80 historických pozorovaní



Obr. D.4: Histogram predikčných chýb modelu MA(30) na linke Liberec Praha s granularitou 1 záznam za 20 minút, predpovede na 7 období dopredu, pri predpovediach bolo braných do úvahy 40 historických pozorovaní



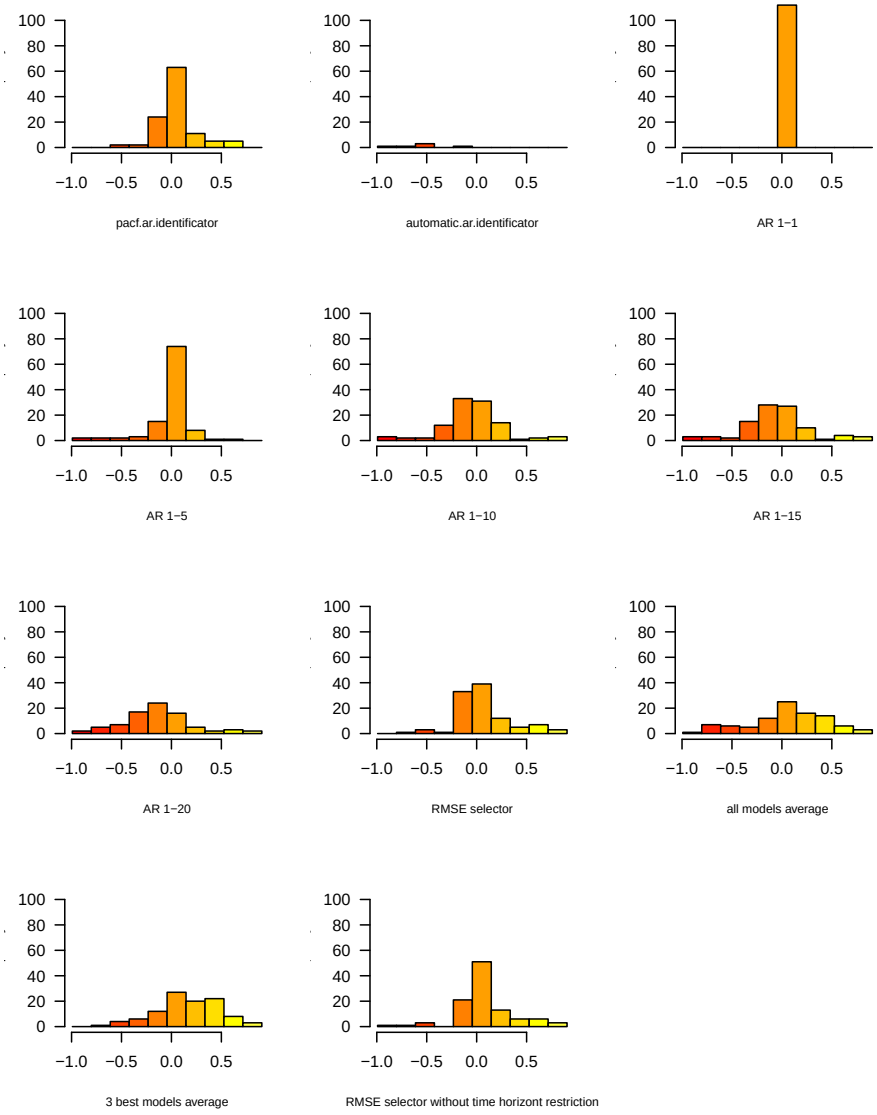
Obr. D.5: Histogram predikčných chýb modelu MA(30) na linke Praha Hradec Králové s granularitou 1 záznam za 12 hodín, predpovede na 7 období dopredu, pri predpovediach bolo braných do úvahy 60 historických pozorovaní



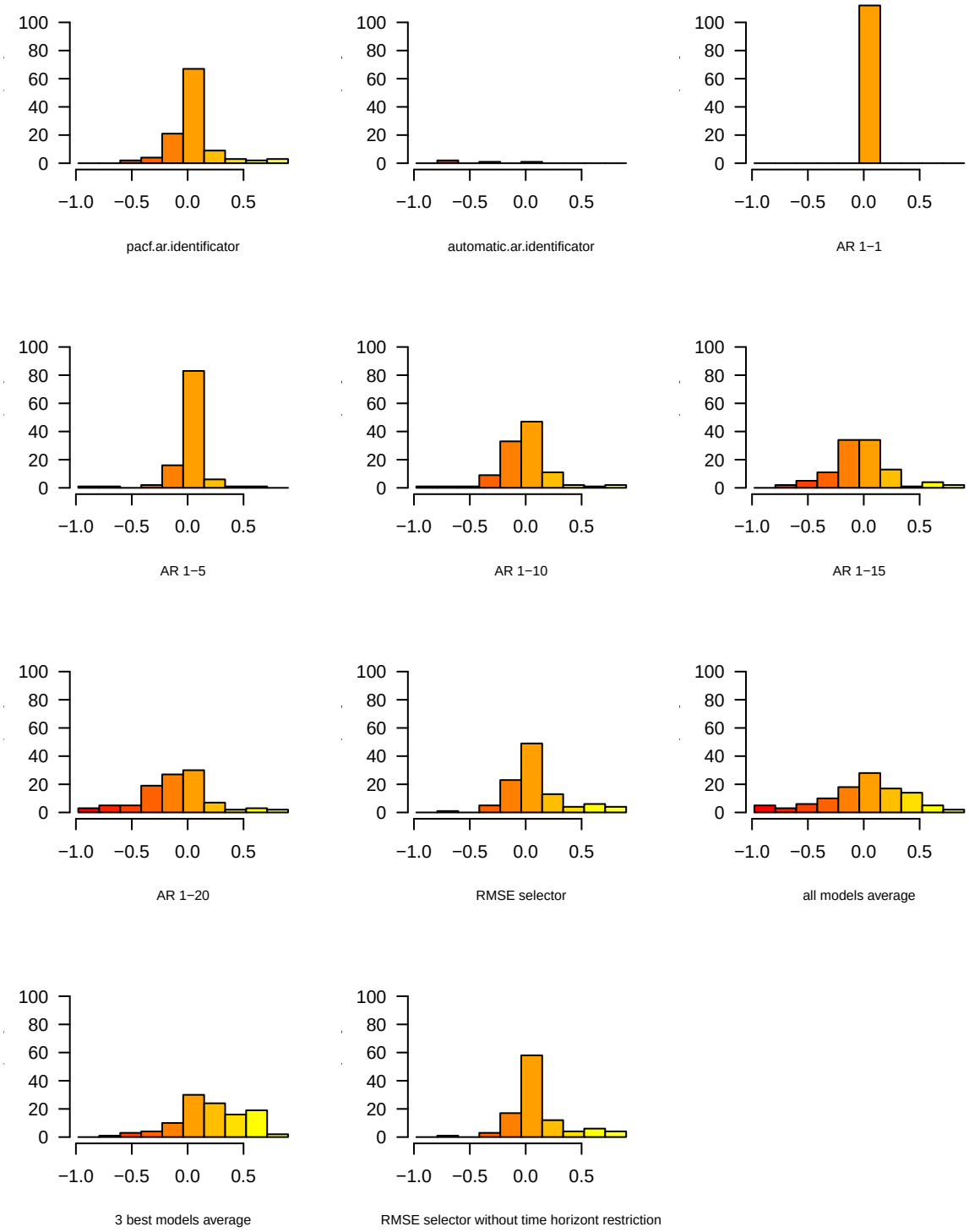
Obr. D.6: Histogram predikčných chýb modelu MA(30) na linke Praha Internet s granularitou 1 záznam za 12 hodín, predpovede na 7 období dopredu, pri predpovediach bolo braných do úvahy 60 historických pozorovaní

Dodatok E

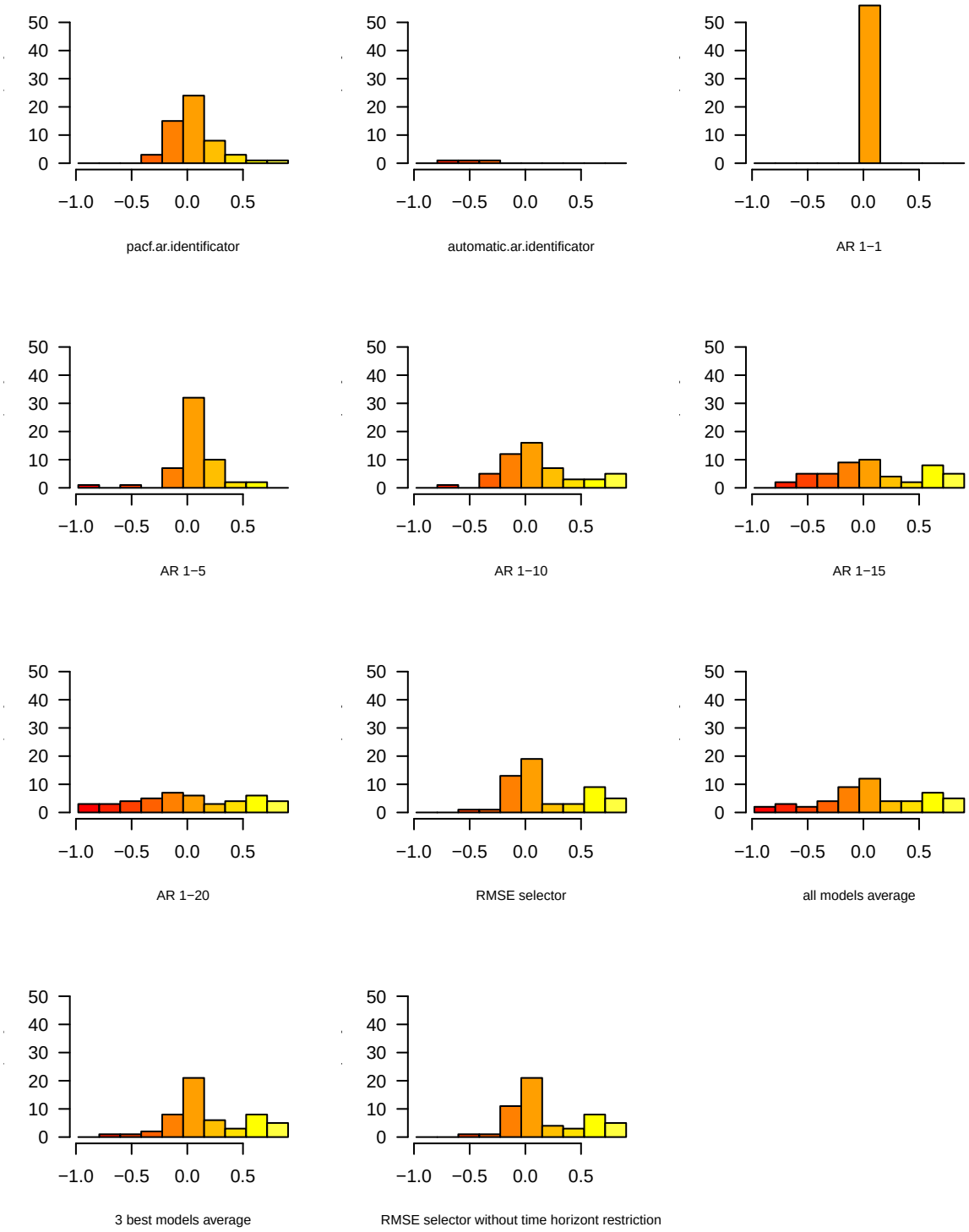
Analýza citlivosti parametrov algoritmov identifikujúcich AR model



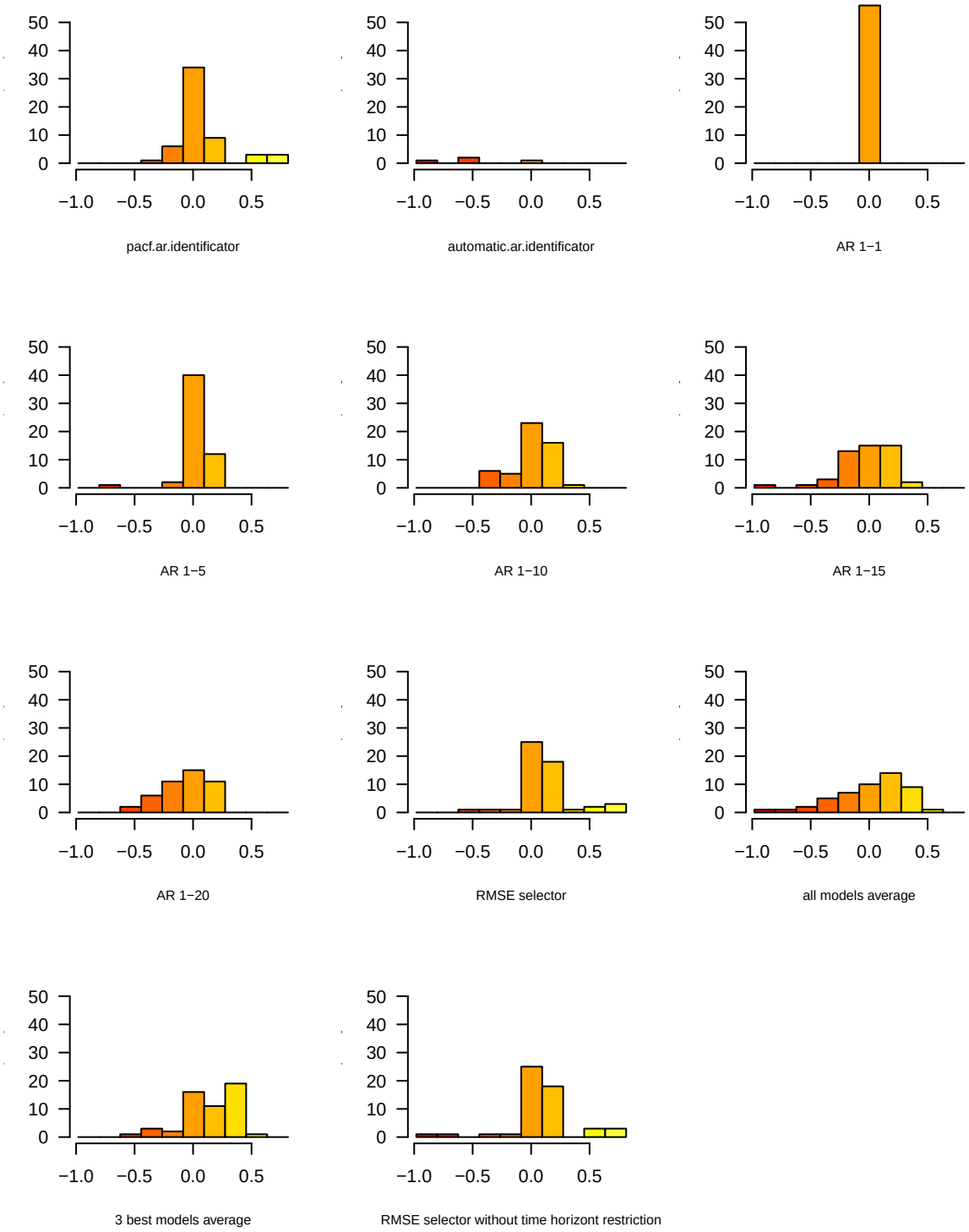
Obr. E.1: RMSE benchmark algoritmov identifikujúcich AR model, pri predikcii bolo použitých 40 historických pozorovaní



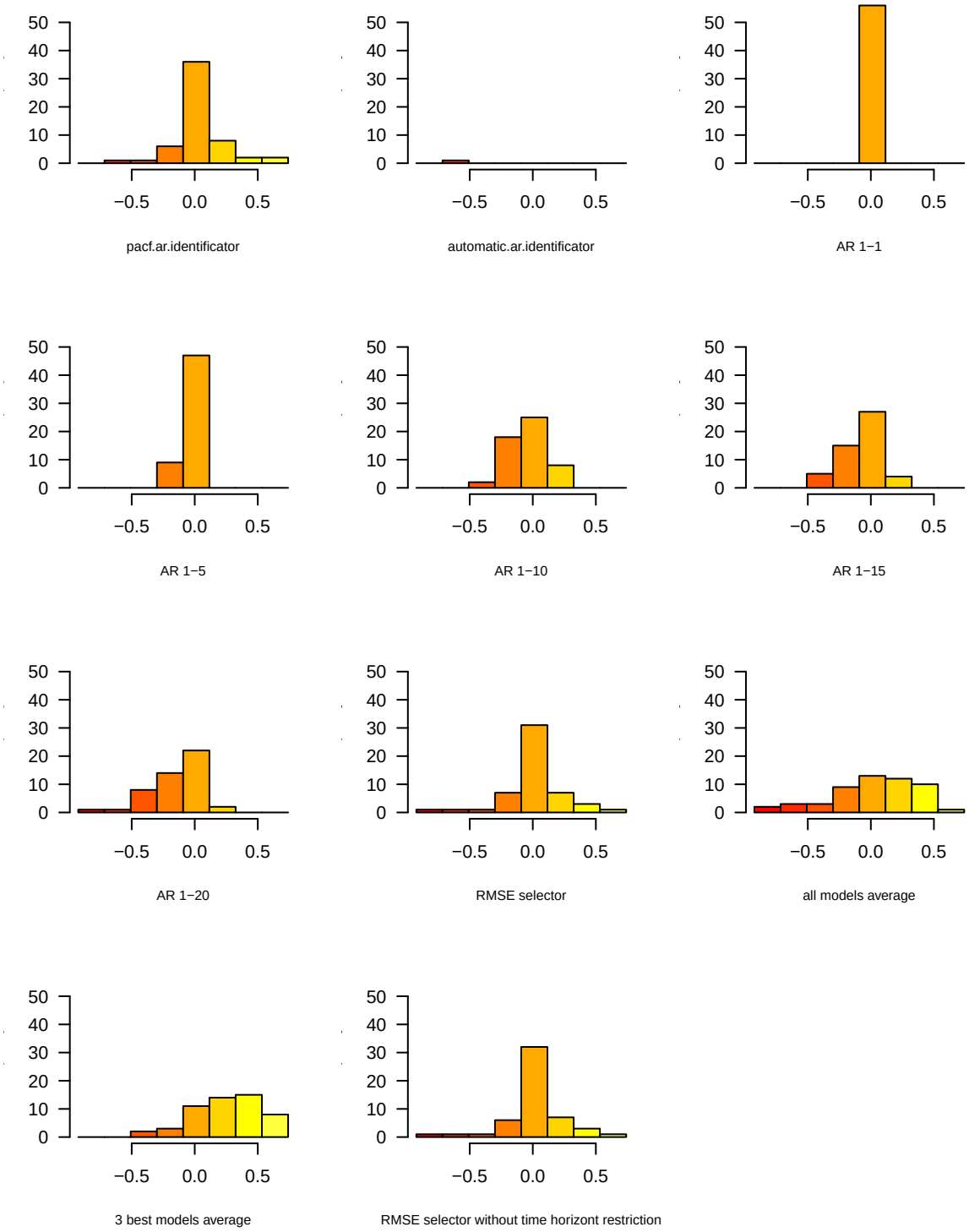
Obr. E.2: RMSE benchmark algoritmov identifikujúcich AR model, pri predikcii bolo použitých 80 historických pozorovaní



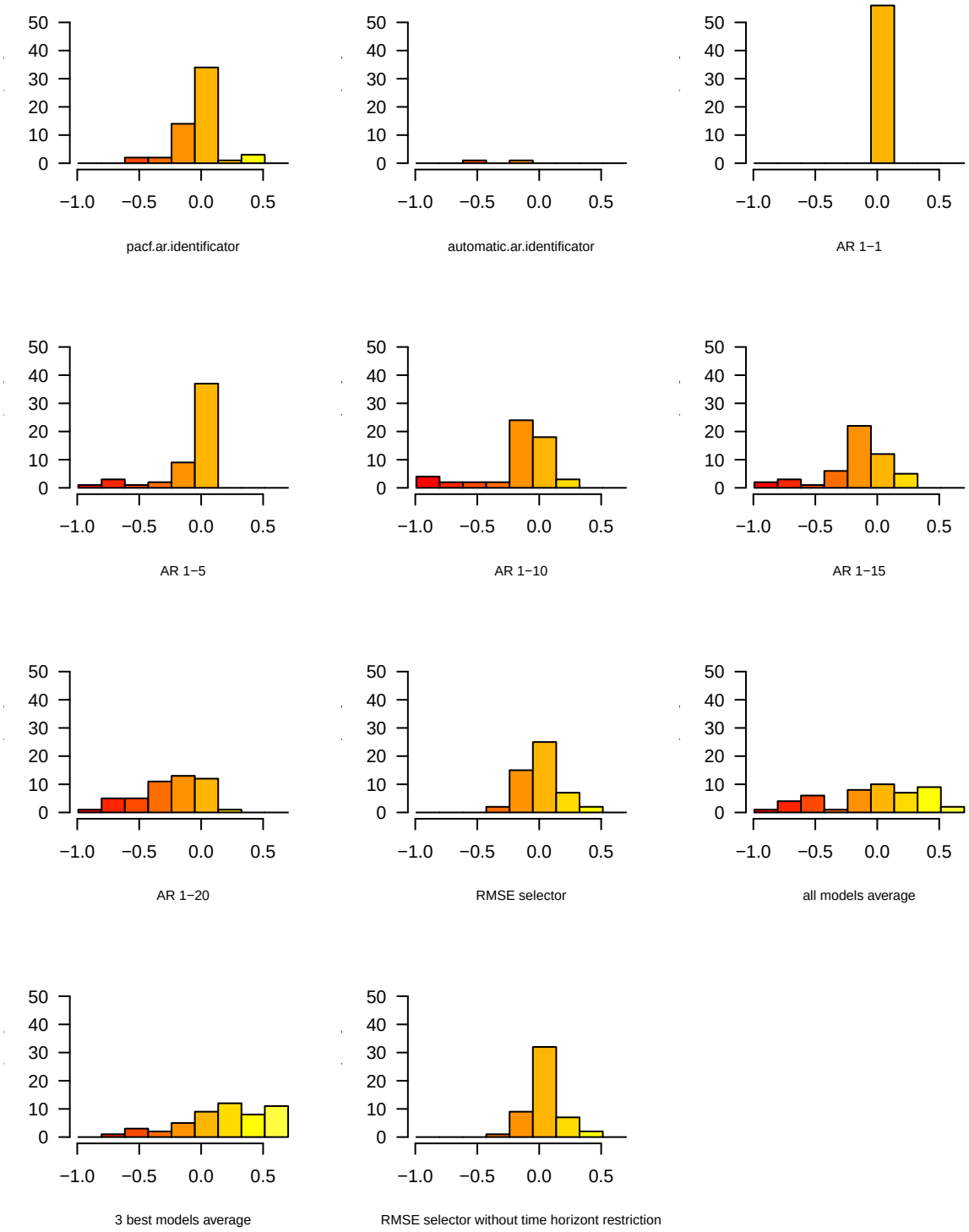
Obr. E.3: RMSE benchmark algoritmov identifikujúcich AR model, pri predikcii bolo použitých 40 a 80 historických pozorovaní, predpoveď na 1 obdobie dopredu



Obr. E.4: RMSE benchmark algoritmov identifikujúcich AR model, pri predikcii bolo použitých 40 a 80 historických pozorovaní, predpoveď na 3 obdobie dopredu



Obr. E.5: RMSE benchmark algoritmov identifikujúcich AR model, pri predikcii bolo použitých 40 a 80 historických pozorovaní, predpoveď na 5 obdobie dopredu



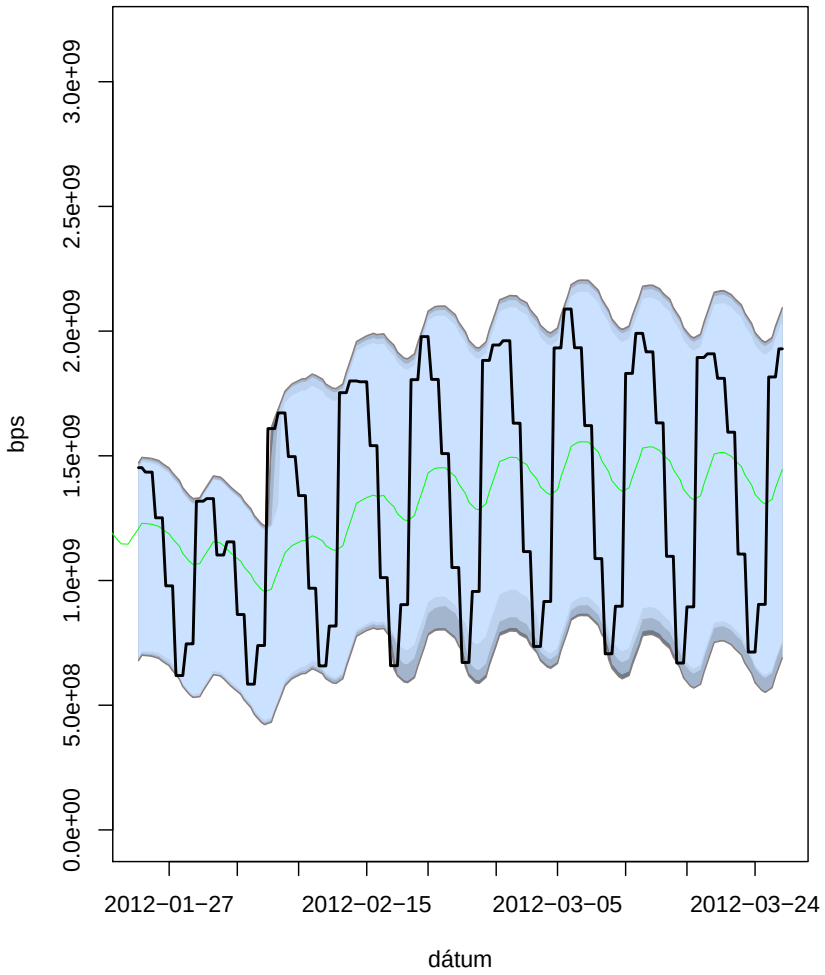
Obr. E.6: RMSE benchmark algoritmov identifikujúcich AR model, pri predikcii bolo použitých 40 a 80 historických pozorovaní, predpoveď na 7 obdobie dopredu

Dodatok F

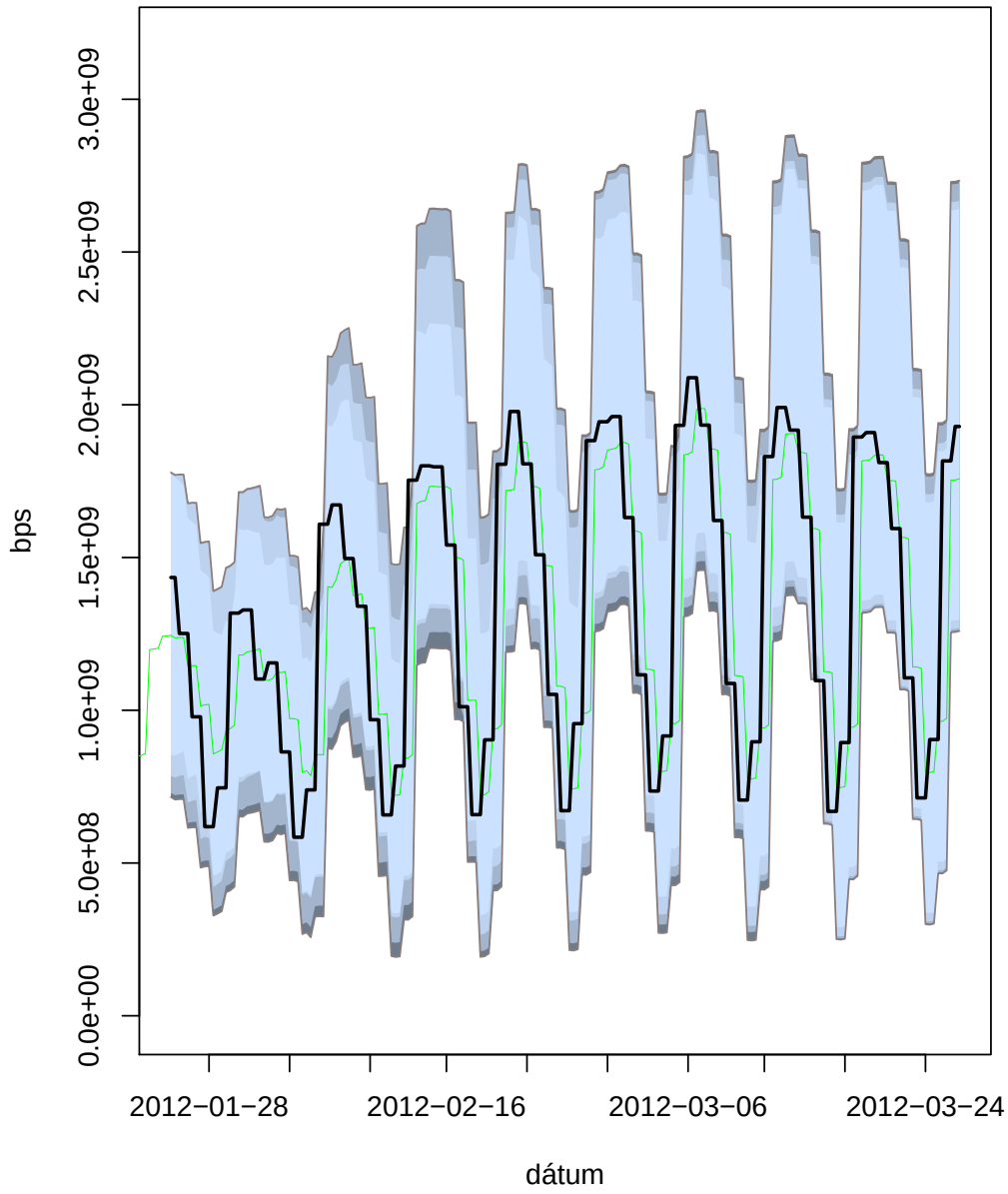
Simulácia predikovania predikčných intervalov

V nasledujúcich grafoch sú 90%, 95%, 99% a 100% predikčné intervaly vykreslené od svetlomodrej až po tmavomodrú farbu. Čierna krivka je skutočná hodnota dátovej linky, zelená je predikcia pomocou modelu.

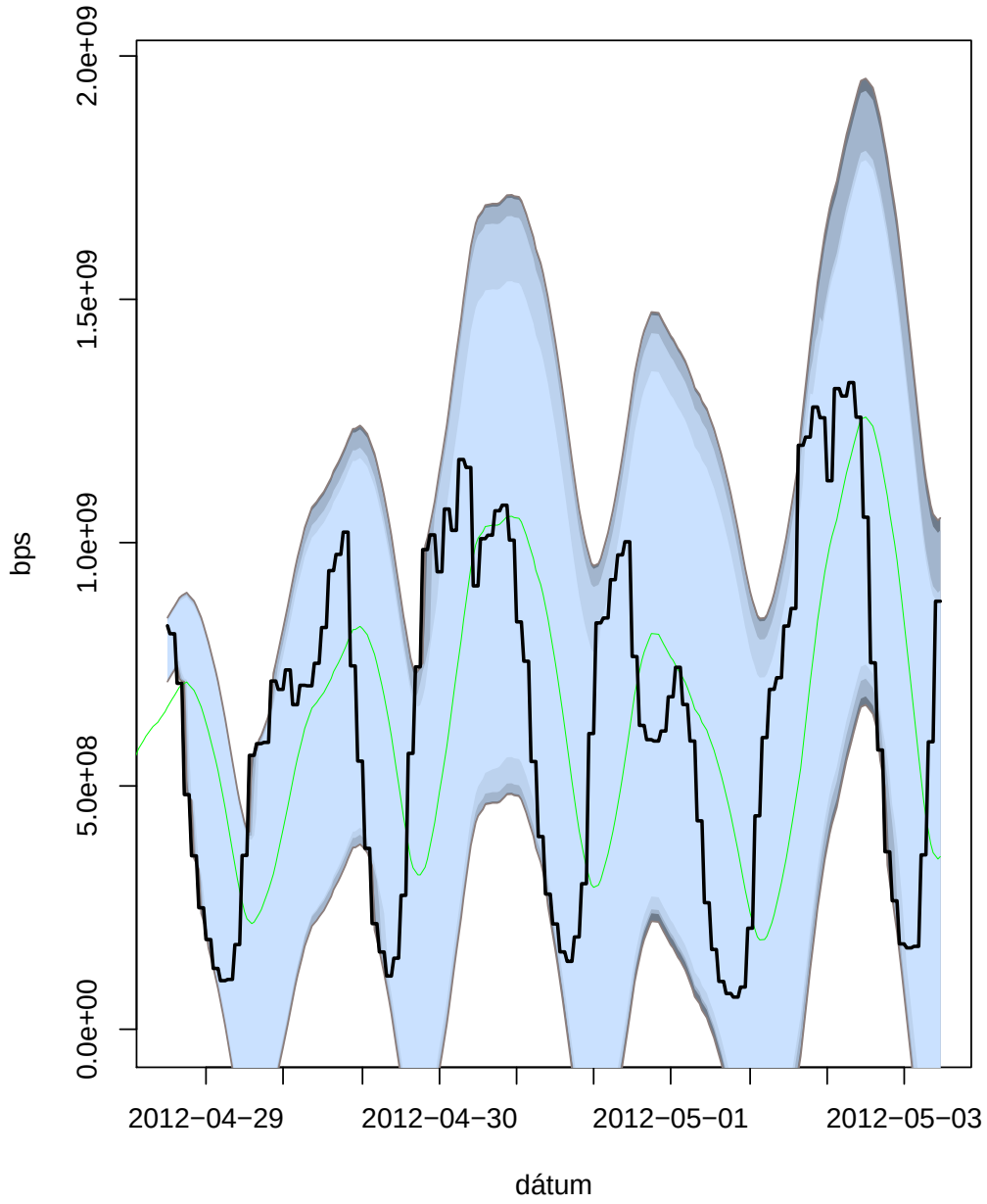
F.1 Predikcia na jedno obdobie dopredu



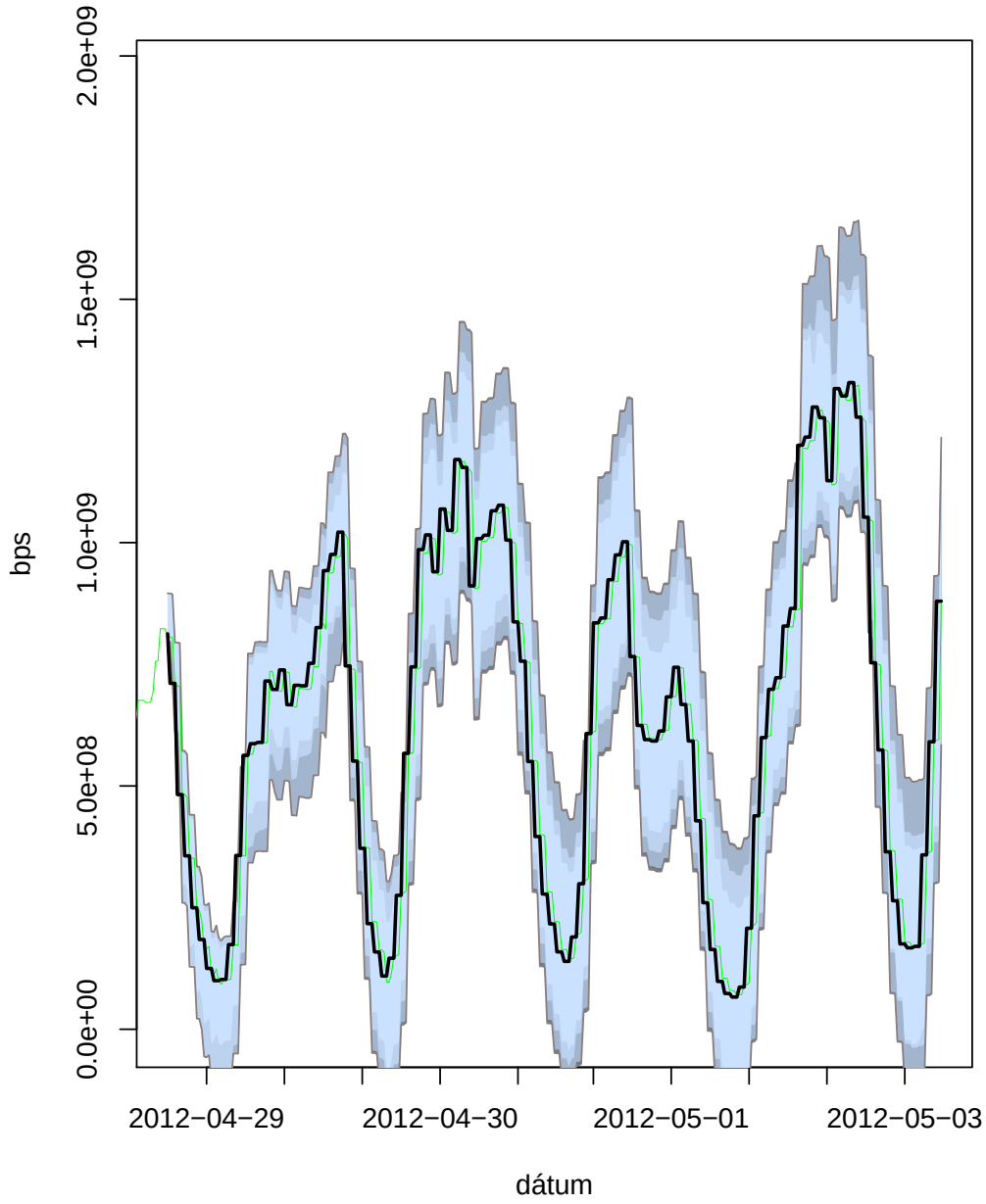
Obr. F.1: Predikčné intervaly modelu MA(25) na linke Praha Hradec Králové, granularita dát 1 záznam za 8 hodín, predpoveď jedného nasledujúceho obdobia.



Obr. F.2: Predikčné intervaly algoritmu vyberajúceho zo všetkých dostupných modelov najlepší model na základe RMSE na linke Praha Hradec Králové, granularita dát 1 záznam za 8 hodín, predpoveď jedného nasledujúceho obdobia.

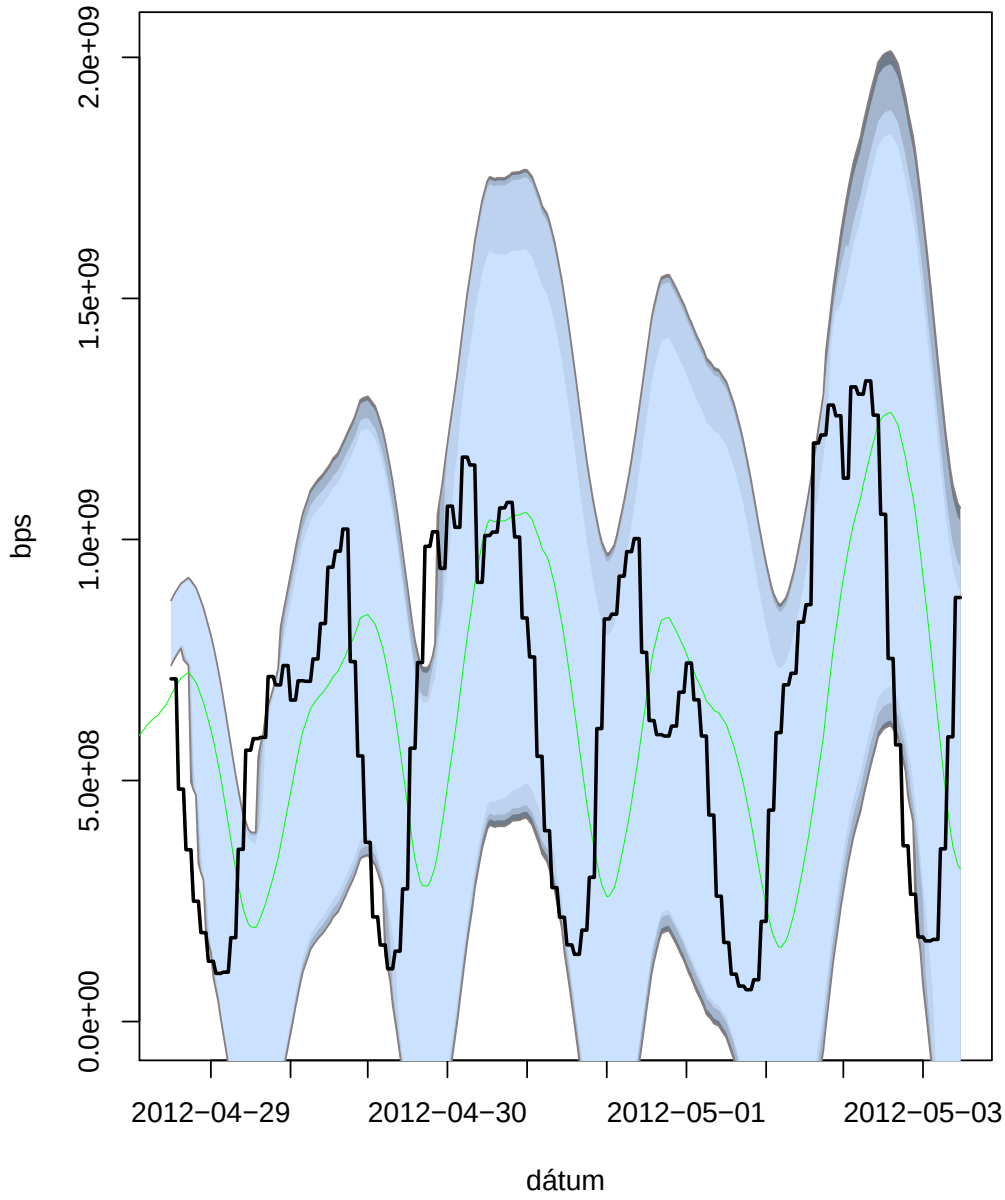


Obr. F.3: Predikčné intervaly modelu MA(25) na linke AMS-IX Praha, granularita dát 1 záznam za 20 minút, predpoveď jedného nasledujúceho obdobia.

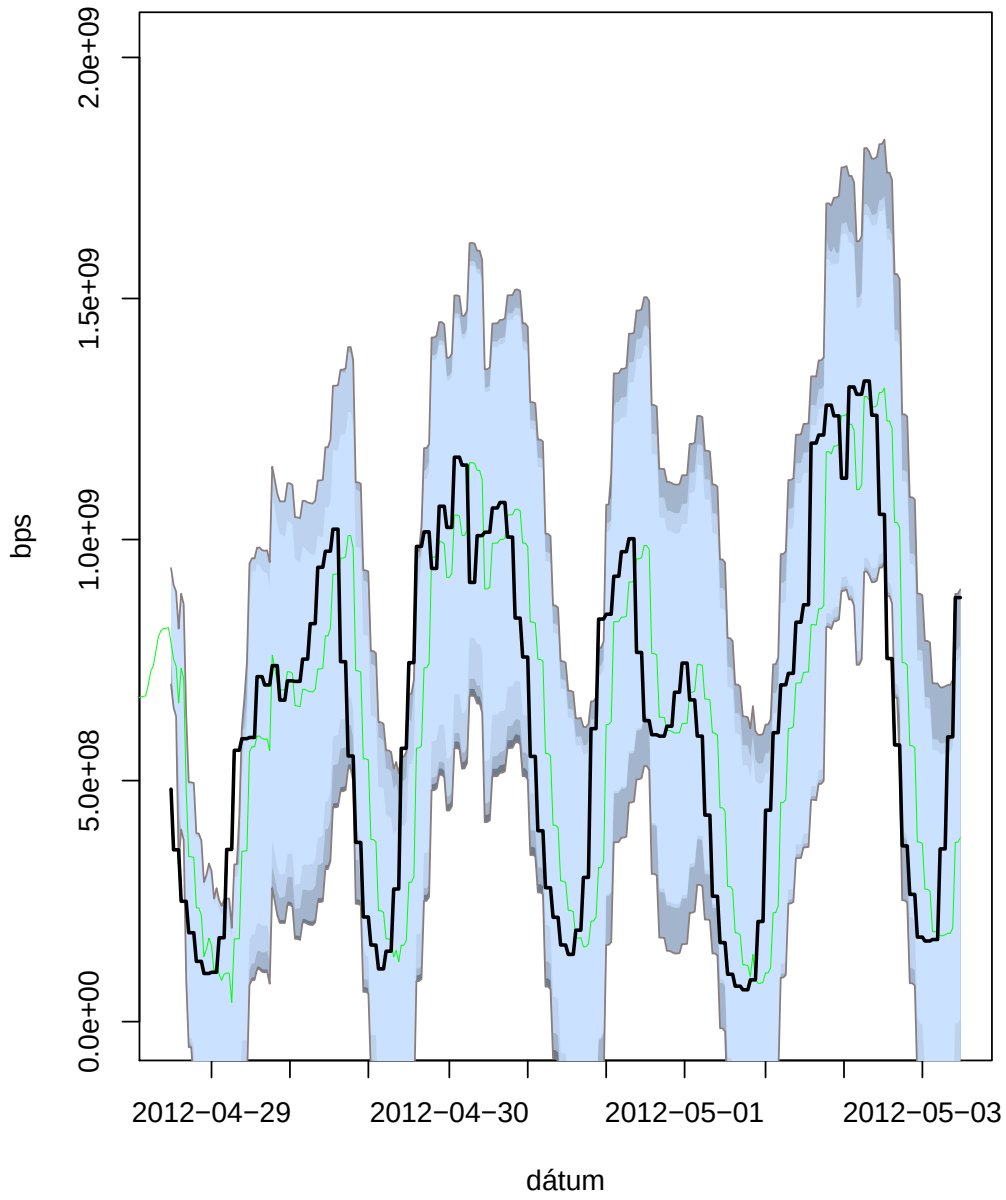


Obr. F.4: Predikčné intervaly algoritmu vyberajúceho zo všetkých dostupných modelov najlepší model na základe RMSE na linke AMS-IX Praha, granularita dát 1 záznam za 20 minút, predpoveď jedného nasledujúceho obdobia.

F.2 Predikcia o päť období dopredu

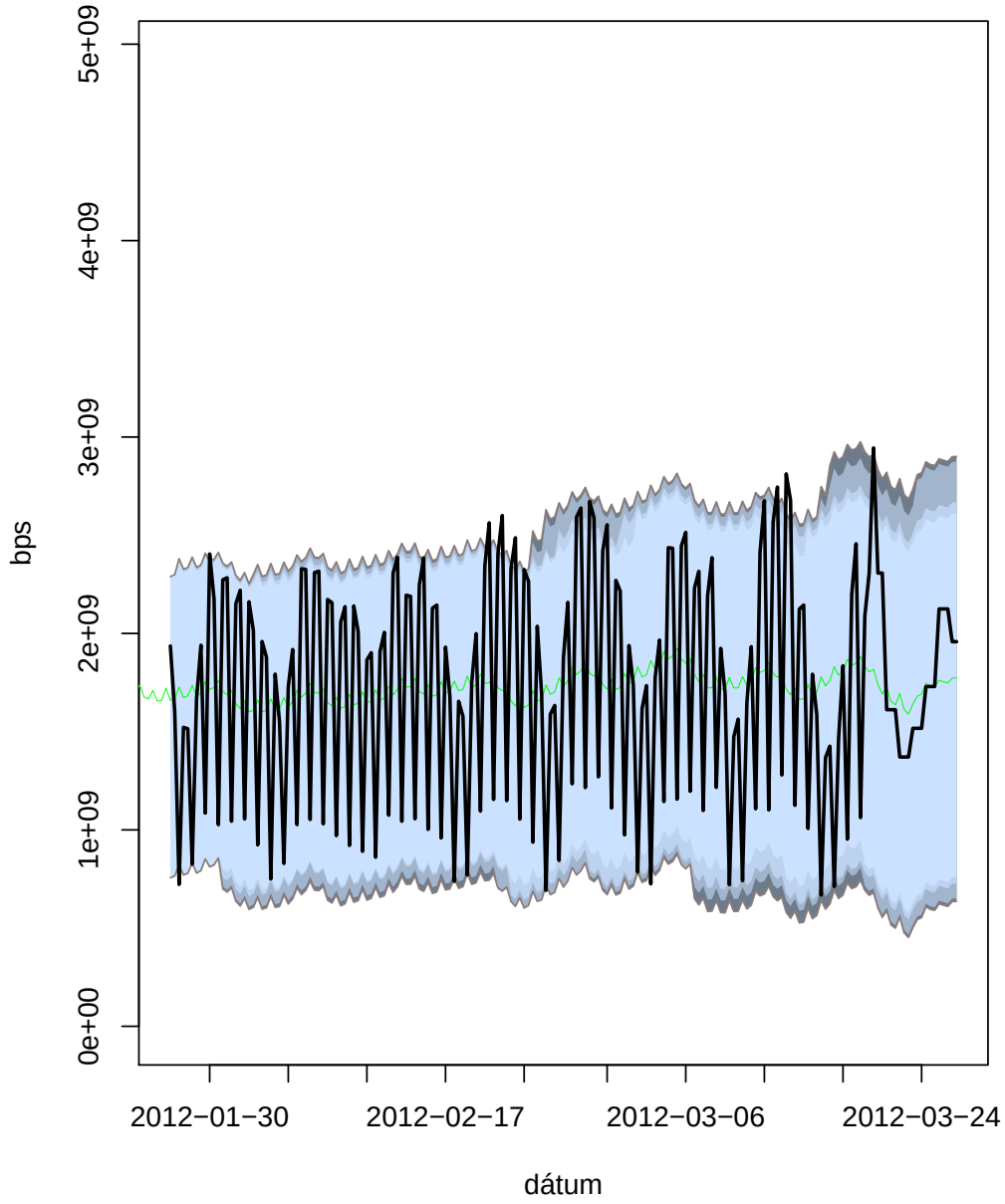


Obr. F.5: Predikčné intervaly modelu MA(25) na linke AMS-IX Praha, granularita dát 1 záznam za 20 minút, predpoveď na 5. budúce obdobie (v tomto prípade 2 hodín dopredu).

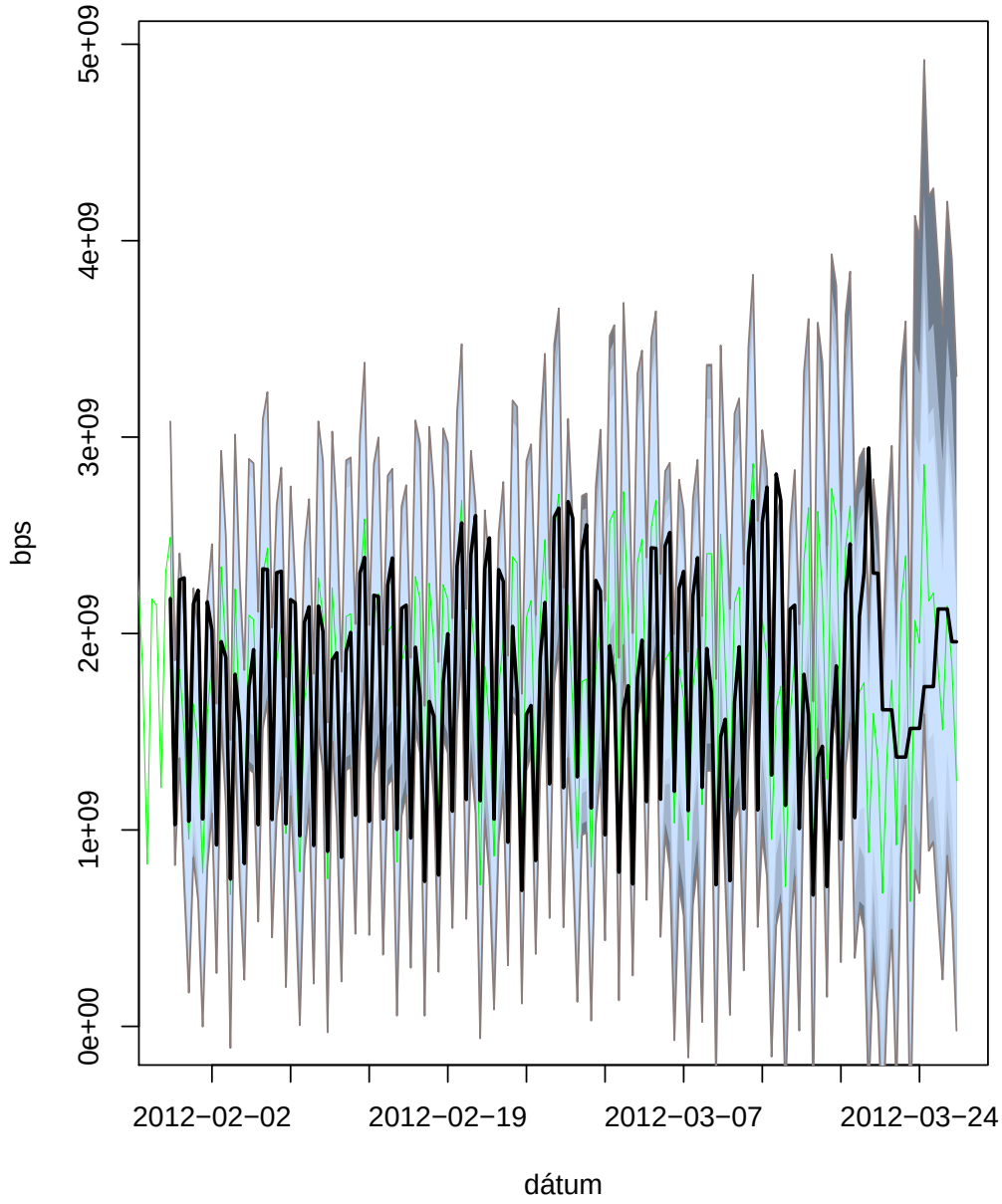


Obr. F.6: Predikčné intervaly algoritmu vyberajúceho zo všetkých dostupných modelov najlepší model na základe RMSE na linke AMS-IX Praha, granularita dát 1 záznam za 20 minút, predpoveď na 5. budúce obdobie (v tomto prípade 2 hodín dopredu).

F.3 Predikcia o desať období dopredu



Obr. F.7: Predikčné intervaly modelu MA(25) na linke Praha NIX, granularita dát 1 záznam za 8 hodín, predpoveď na 10. budúce obdobie (v tomto prípade 80 hodín dopredu).



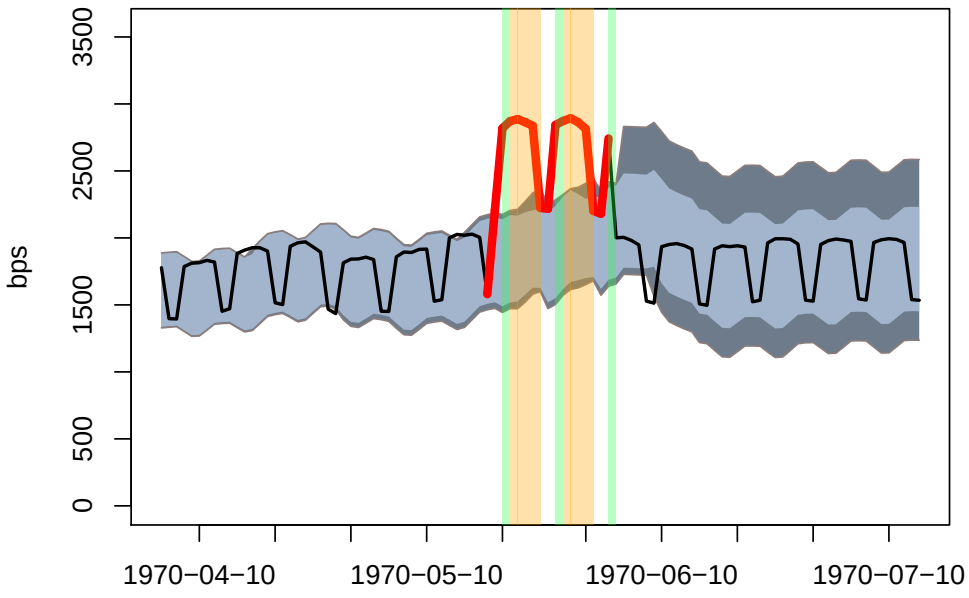
Obr. F.8: Predikčné intervaly algoritmu vyberajúceho zo všetkých dostupných modelov najlepší model na základe RMSE na linke Praha NIX, granularita dát 1 záznam za 8 hodín, predpoveď na 10. budúce obdobie (v tomto prípade 80 hodín dopredu).

Dodatok G

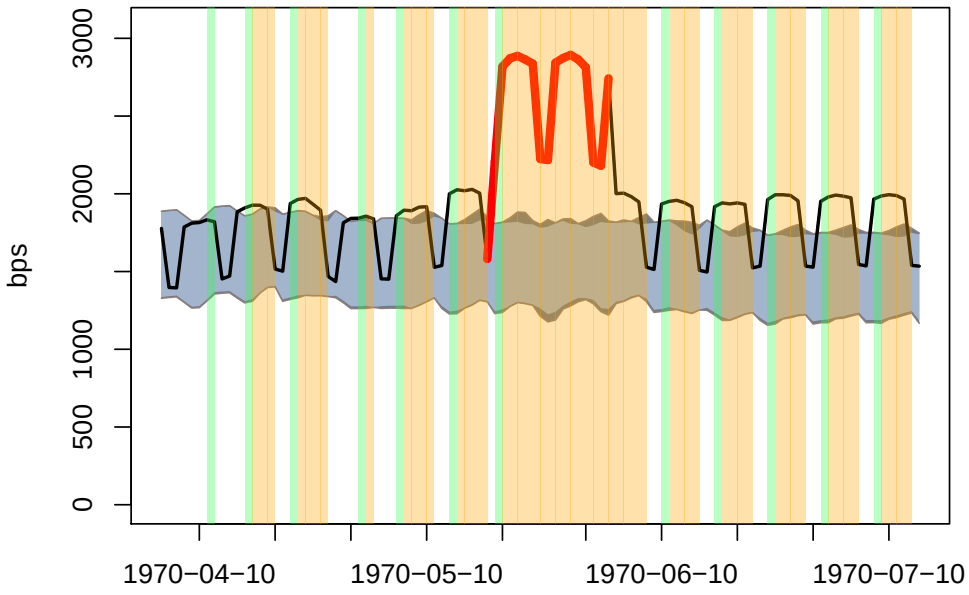
Simulácia detekovania anomálií

Táto kapitola obsahuje výsledky simulácie detekovania anomálií. Grafy časových radov detekovania anomálií obsahujú dátové toky bez anomálií (krivky čiernej farby), skutočné anomálie (krivky červenej farby), detekované možné anomálie (zelený interval) a detekované anomálie (oranžový interval).

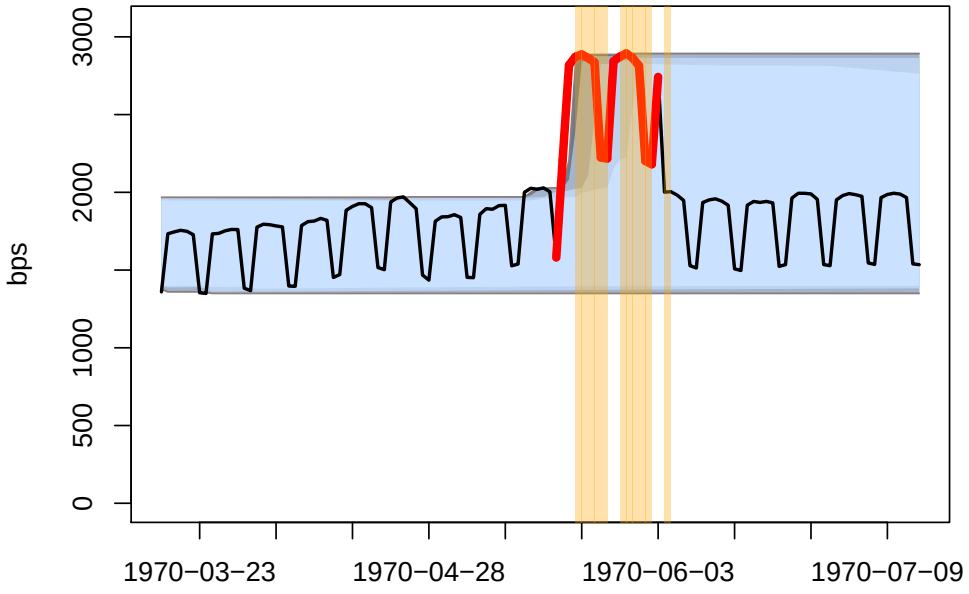
Tam, kde sa je v jednom okamžiku skutočná anomália (červená krivka) a detekovaná anomália (oranžový interval) bola pozitívne detekovaná anomália (true positive). V miestach, kde je oranžový interval na čiernej krivke došlo k nesprávne detekovanej anomálii (false positive).



Obr. G.1: Detekcia anomálií na simulovaných sezónnych dátach, dvojstavový algoritmus využívajúci predikčné intervaly skonštruované pomocou algoritmu vyberajúceho najlepší model, senzitivita 0,6



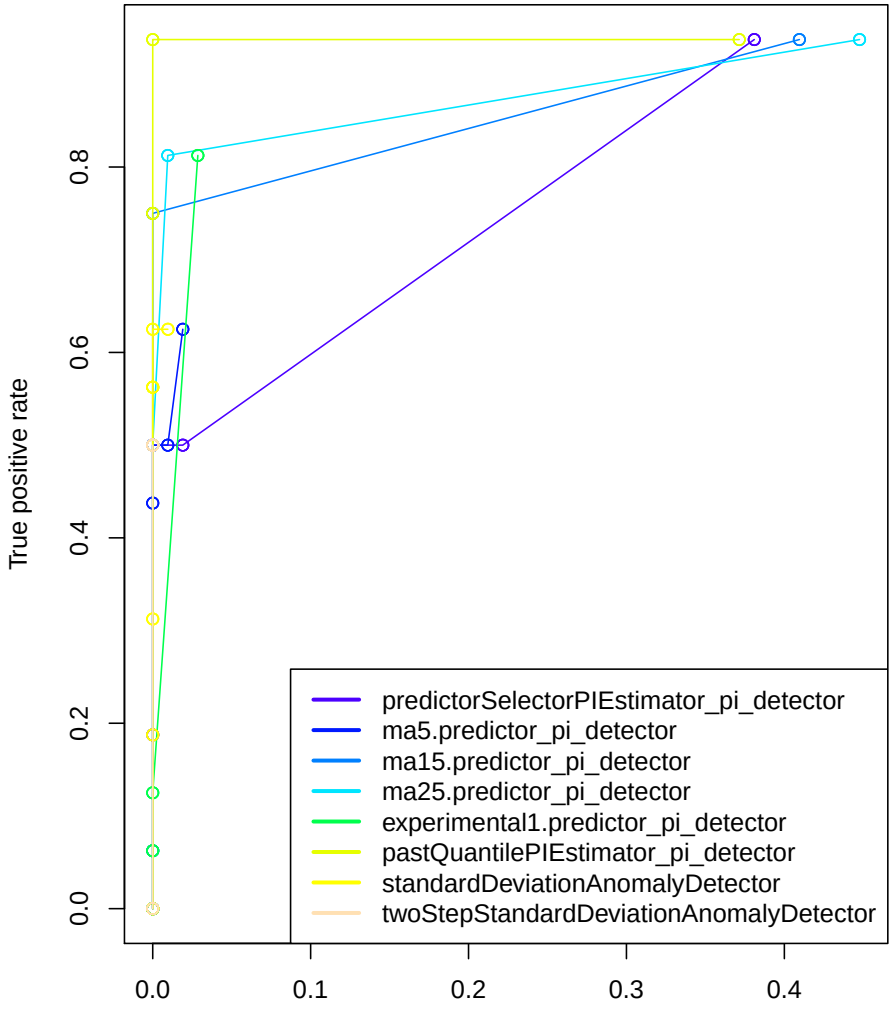
Obr. G.2: Detekcia anomálií na simulovaných sezónnych dátach, dvojstavový algoritmus využívajúci predikčné intervaly skonštruované pomocou algoritmu vyberajúceho najlepší model, senzitivita 1



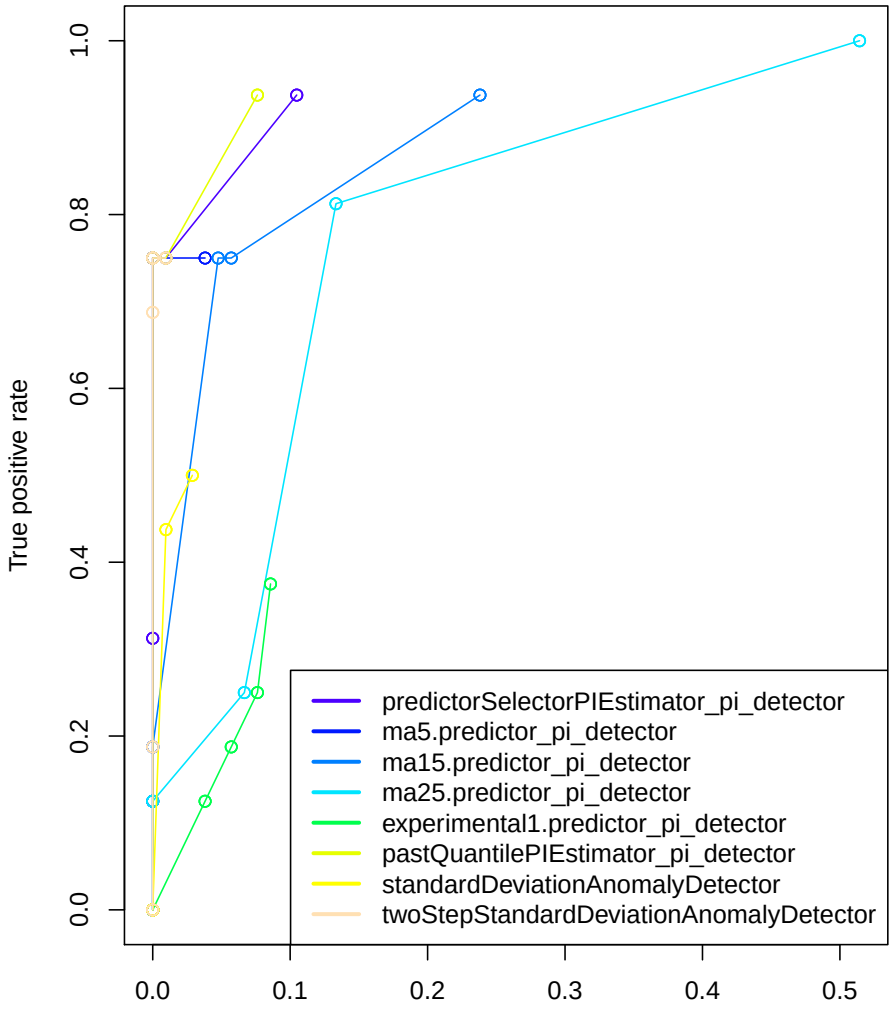
Obr. G.3: Detekcia anomálií na simulovaných sezónnych dátach, primitívny algoritmus detekujúci anomálie na základe 3σ, senzitivita 1

G.1 ROC krivky detektorov anomálií

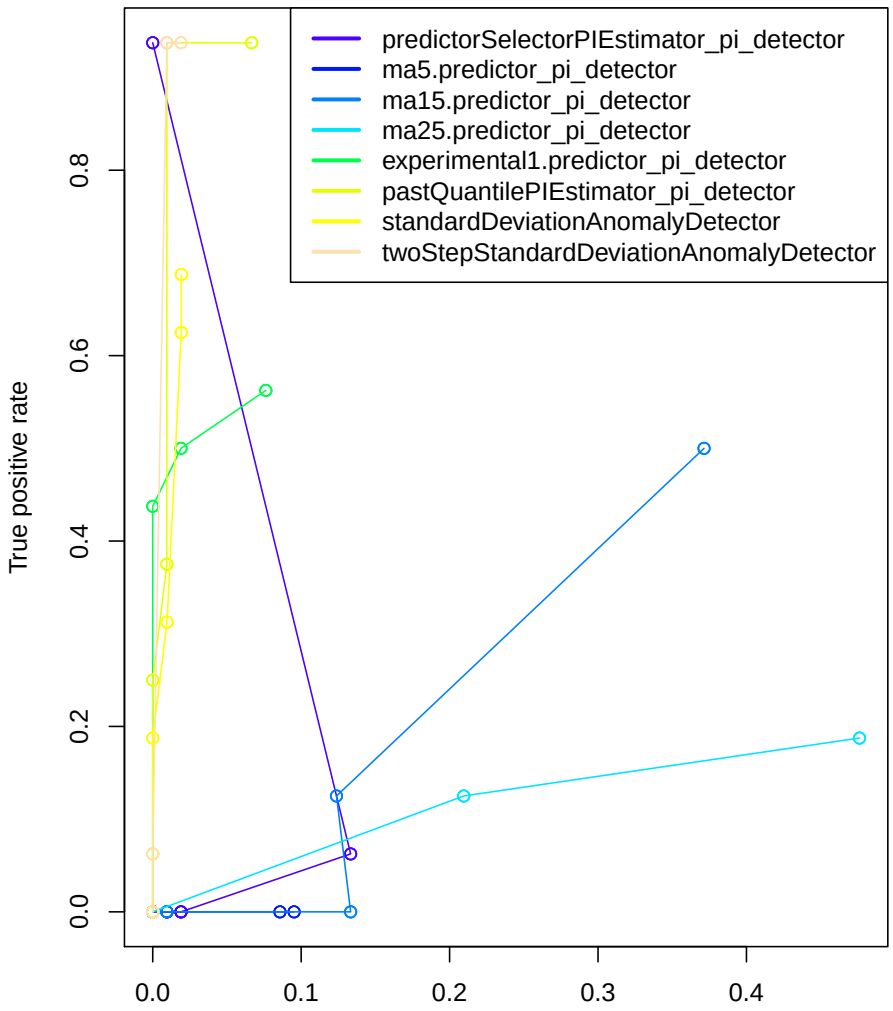
Grafy v tejto kapitole vykresľujú ROC krivky detektorov anomálií. Na osy X je početnosť falošne pozitívnych prípadov (počet falošne detekovaných prípadov/počet normálnych stavov), na osy Y je početnosť skutočne pozitívnych prípadov (počet správne detekovaných anomálií/celkový počet anomálií). Krivka je vykreslená pre daný detektor a rôzne konfigurácie jeho parametru citlivosti (jednotlivé body predstavujú samotné simulácie s daným parametrom citlivosti). Pre detektor platí, že lepšia je vyššia hodnota na osy Y a nižšia na osy X.



Obr. G.4: ROC krivky detekcie anomálií na simulovaných dátach s granularitou 1 záznam za deň



Obr. G.5: ROC krivky detekcie anomálií na simulovaných dátach s granularitou 1 záznam za 2 hodiny



Obr. G.6: ROC krivky detekcie anomálií na simulovaných dátach s granularitou 1 záznam za 2 hodiny

Dodatok H

Simulátor predikcií

Pre praktické overenie predikčných modelov sme sa rozhodli implementovať všetky modely uvedené v kapitole 4 a simulovať predikcie pomocou týchto modelov. Modely sme implementovali v štatistickom prostredí R[1]. Simulácie sme spúšťali pomocou nami implementovaného simulátora, ktorý sekvenčne simuloval na časových radoch predikcie a následne ich vyhodnocoval.

Štatistické prostredie R síce obsahuje základné nástroje pre simulovanie predikcií na časových radoch, tieto nástroje ale poskytujú len základné možnosti predikcií [45]. Neobsahujú napríklad možnosť vyhodnotenia výkonnosti jednotlivých modelov, neumožňujú simulovanie predikcií predikčných intervalov a simulovanie detekcií anomálií.

Výhodou nami implementovaného simulátora je mimo iné jednoduchá rozšíriteľnosť o ďalšie predikčné modely, možnosť analýzy výkonnosti jednotlivých algoritmov a možnosť spúšťania predikčných algoritmov v multiprocesorovom prostredí, prípadne na gridoch.

H.1 Architektúra simulátora

Vzhľadom k obmedzenej možnosti objektového programovania v štatistickom prostredí R [30] sme rozdelili jednotlivé časti simulátora do modelov, ktoré odpovedajú ich funkcionalite. V nasledujúcom texte popíšeme podrobnejšie jednotlivé moduly.

- *Experiment* obsahuje konfiguráciu spúšťaného experimentu. Obsahuje teda zoznam dátových sád, na ktorých bude simulovaná predikcia, zoznam predikčných modelov a tiež parametre pre simuláciu (veľkosť pamäte pre historické pozorovania, časový horizont predpovede a iné). Taktiež obsahuje časť, ktorá generuje štatistické výsledky a grafy z experimentu. Táto časť je implementovaná ako funkcia *generateResults*.
- *ExperimentExecutor* úlohou modulu typu ExperimentExecutor je spúšťanie jednotlivých simulácií z modulu experiment a následné generovanie výsledkov. Súčasná implemntácia obsahuje nasledujúce tri moduly typu *ExperimentExecutor*.
  - *SingleProcessorExperimentExecutor* spúšťa simulácie na jednom procesore sekvenčne. Po dokončení jednej simulácie je spustená ďalšia simulácia.
  - *MultiprocessorExperimentExecutor* spúšťa viacero simulácií paralelne na všetkých dostupných procesoroch počítača. Po dokončení simulácií je spustená generácia výsledkov experimentu (tá beží len na jednom procesore). Pre spustenie experimentov pomocou tohto modulu je nutné, aby bol v štatistickom prostredí R prítomný prídavný modul parallel[46].

1. http://www.r-project.org/

  - *GridExperimentExecutor* spúšťa experiment v prostredí gridu, pričom každá simulácia je spustená ako samostatná úloha. Všetky simulácie z daného experimentu môžu teda bežať paralelne. Generácia výsledkov je spustená ako samostatná úloha, ktorá je spustená po dokončení všetkých simulácií. *GridExperimentExecutor* beží v prostredí gridu Metacentrum, pomocou niekoľkých úprav je možné spúšťať simulácie aj na iných gridoch. Jediná požiadavka je nainštalované štatistické prostredie R v gride.

- *ResultCollector* slúži ako objekt zhromažďujúci výsledky predikcie. Pri bodovej predikcii je to predovšetkým samotná predikcia. Pri simulácii predikcie predikčných intervalov sú to predikčné intervaly pre vopred nastavené kvantily a pre anomálie sú to stavy detekcie anomálií. Taktiež zhromažďuje informácie o výpočtovej náročnosti modelov. Tento modul je možné priamo v simulácií dynamicky rozširovať.

- *ResultGenerator* je modul obsahujúci funkcionalitu pre zhromažďovanie štatistických údajov o výsledku experimentu. Je využívaný nasledujúcimi modulmi:

  - *GraphResultGenerator* slúži na generovanie grafov pomocou nástroja GnuPlot[23]. V súčasnej implementácii obsahuje funkcionalitu pre generovanie grafov predikcií časových radov, reziduí predikcií, predikčných intervalov, grafov s anomáliami a ROC krivkami. Formát generovaných grafov sa nastavuje v module *Experiment*.
  - *StatisticsResultGenerator* obsahuje implementáciu funkcií generujúcich metriky kvality predpovedí ako RMSE, MAPE, Theil's U a koeficienty pre ROC krivky. Tieto štatistiky sú zhromažďované v module Experiment a následne ukladané do csv tabuliek.

- *Predictor* je modul vytvárajúci predikcie na základe historických dát. Moduly typu prediktor obsahujú všetky popísané modely v práci (lineárne regresné, MA modely a ďalšie). Súčasťou implementácie v module Prediktor sú aj funkcie pre automatickú detekciu modelu. Pre implementáciu nových predikčných modelov je nutné vytvoriť nový modul typu prediktor. Vstupom pre tento modul sú historické časové rady, výstupom je samotná predikcia.

- *ExperimentData* je modul zapúzdrujúci načítanie dát pre simuláciu. V implementácii priloženej k tejto práci je modul načítavajúci dáta formátu RDX, v ktorom sú uložené historické dátové toky siete CESNET. Ďalej obsahuje súčasná implementácia moduly pre vytváranie simulovaných dát popísaných v kapitole 3.2. Súčasná implementácia taktiež umožňuje načítavať a simulovať predikcie na historických časových radoch ekonomického charakteru z databáze FRED (Federálny rezervný systém, pobočka v Saint Louisiane)[15].

H.2 Vytváranie vlastných experimentov

Pre vytvorenie vlastných dát je nutné vytvoriť modul typu *Experiment* a následne ho spustiť pomocou modulu *ExperimentExecutor*. Ako najjednoduchšie považujeme skopírovať jeden z modulov dostupných v súčasnej implementácii a upraviť ho k vlastným potrebám. Konfigurácia simulovaných dát sa nachádza v premennej *experimentData*. Pre vlastné dáta bude nutné vytvoriť nový modul typu *ExperimentData*, ktorý načítava dáta.

H.3 Možnosť rozšírenia simulátora o vlastné predikčné modely

Pre rozšírenie modelu o nové predikčné modely pre bodovú alebo intervalovú predikciu je nutné vytvoriť nový modul typu *Prediktor*. Pre simulovanie iných typov predikcií je nutné ešte vytvoriť nový modul *ExperimentExecutor*. Modul *ResultCollector* pravdepodobne nebude treba modifikovať, je možné ho dynamicky rozširovať o nové atribúty simulácie bez nutnosti zmeny jeho kódu.

Dodatok I

Zoznam elektronických príloh

- cesnet_data – zdrojové dáta zo siete CESNET2, na ktorých boli vykonávané experimenty
- experiment_results – výsledky experimentov v neagregovanej forme
  - anomalies_detection – simulácie detekovania anomálií
  - prediction_intervals – simulácie intervalových predpovedí
  - prediction_methods – simulácie bodových predpovedí
- prediction_simulator – simulátor predikcií napísaný v štatistickom prostredí R