



UNIVERZITA KOMENSKÉHO, BRATISLAVA
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

VIZUALIZÁCIA VERIFIKÁCIE PREDPOVEDNÝCH MODELOV POČASIA

Diplomová práca

Bratislava, 2015
Bc. Marek Kružliak



UNIVERZITA KOMENSKÉHO, BRATISLAVA
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

VIZUALIZÁCIA VERIFIKÁCIE PREDPOVEDNÝCH MODELOV POČASIA

Diplomová práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: RNDr. Andrej Lúčny, PhD.

Bratislava, 2015
Bc. Marek Kružliak



ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Marek Kružliak

Študijný program: aplikovaná informatika (Jednoodborové štúdium,
magisterský II. st., denná forma)

Študijný odbor: 9.2.9. aplikovaná informatika

Typ záverečnej práce: diplomová

Jazyk záverečnej práce: slovenský

Názov: Vizualizácia verifikácie predpovedných modelov počasia / *Visualization of forecast models verification*

Ciel: Cieľom práce je návrh a implementácia techník vizualizácie výsledkov verifikácie numerických predpovedí počasia. Dôraz je kladený na kompaktné zobrazenie týchto mnohorozmerných dát pomocou 2D obrázka (prípadne aj s interakciou), a rýchlu identifikáciu trendov a vzorov, za účelom odhalenia chýb nastavenia predpovedného modelu. Práca bude zahŕňať taktiež implementáciu samotnej verifikácie výsledkom konkrétneho predpovedného modelu (WRF).

Literatúra:

1. Lundblad, P. Lofving, H. Elofsson, A. Johansson, J.: Exploratory Visualization for Weather Data Verification
2. Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Duda, M., Huang, X.-Y., Wang, W., Powers, J. G., 2008: A Description of the Advanced Research WRF Version 3, NCAR Technical Note.
3. Mason, S. J., 2008: Understanding forecast verification statistics, Meteorological Applications 15: 31-40

Poznámka: implementácia v jazyku Java, mapovanie mnohorozmerných dát do 2D obrázka

Klúčové slová: vizualizácia dát, vizualizácia informácií, verifikácia predpovede počasia, predpovedný model počasia

Vedúci: RNDr. Andrej Lúčny, PhD.

Katedra: FMFI.KAI - Katedra aplikovanej informatiky

Vedúci katedry: doc. PhDr. Ján Rybár, PhD.

Dátum zadania: 27.02.2014

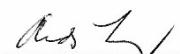
Dátum schválenia: 27.02.2014

prof. RNDr. Roman Ďuríkovič, PhD.

garant študijného programu


.....

študent


.....

vedúci práce

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím citovaných zdrojov.

.....

Ďakujem mnohým ľuďom, ktorí sa zaslúžili o úspešné dokončenie tejto práce. Menovite by som chcel podakovať Mgr. Matejovi Novotnému PhD. za cenné rady pri návrhu vizualizácie, RNDr. Andrejovi Lúčnemu PhD. za vedenie tejto práce, Mgr. Jurajovi Bartokovi PhD. za odborné rady z oblasti meteorológie. Taktiež d'akujem celej svojej rodine a blízkym, obzvlášť mojej snúbenici Janke, ktorá bola vždy pri mne (doslovne), pri písaní tejto práce.

Abstrakt

Verifikácia predpovedných modelov počasia je nepostrádateľnou súčasťou meteorologického výskumu a taktiež celkového procesu predpovedania počasia. Dnešné metódy verifikácie predpovedí počasia sa sústredia najmä na numerický popis výkonu predpovedných modelov, avšak práve na vizualizáciu týchto dát sa kladie nie veľký dôraz. V našej práci sa sústredíme práve na tento aspekt verifikácie a prezentujeme niekoľko techník vizualizácie, ktoré sme navrhli za účelom lepšieho porozumenia dát.

Kľúčové slová: vizualizácia informácií, verifikácia predpovedí počasia

Abstract

Weather forecast model verification is indispensable part of meteorological research and weather forecasting process at all. Modern methods of weather forecast verification are focusing on numerical description of forecast model's performance, yet the visualization of these data is highly underestimated part of this process. In our work we focus on this aspect of verification and we present a few visualization techniques designed for better understanding of data.

Keywords: information visualization, verification of weather forecasts

Obsah

Úvod	1
1 Verifikácia predpovedných modelov počasia	2
1.1 Predpovedný model počasia	2
1.1.1 WRF model	3
1.2 Dáta	3
1.2.1 Predpovedané dáta	4
1.2.2 Pozorované dáta	5
1.2.3 Párovanie dát	5
1.3 Metódy verifikácie	6
1.4 Meranie chyby predpovede	8
1.4.1 Stredná chyba predpovede	8
1.4.2 Stredná absolúttna chyba	9
1.4.3 Stredná kvadratická chyba	9
1.4.4 Všeobecná kumulovaná chyba	9
1.4.5 Medián absolúttnych chýb	11
2 Predchádzajúce riešenia 1	
<i>Verifikačný softvér</i>	12
2.1 Štatistický softvér	12
2.1.1 Tabuľkový softvér	13
2.1.2 MATLAB	13
2.1.3 R	14
2.1.4 SAS	14

2.1.5	IDL	14
2.2	Špecializovaný softvér	15
2.2.1	NCL	15
2.2.2	MET	16
2.2.3	EVS	17
2.3	Zhrnutie	17
3	Predchádzajúce riešenia 2	
	<i>Techniky vizualizácie vo verifikácii</i>	19
3.1	Bodový graf	19
3.1.1	Konštrukcia bodového grafu	19
3.1.2	Kantil-kvantil graf	20
3.1.3	Úloha bodového grafu vo verifikácii	21
3.2	Krabicový diagram	21
3.2.1	Konštrukcia krabicového diagramu	22
3.2.2	Úloha krabicového diagramu vo verifikácii	23
3.3	Histogram	24
3.3.1	Konštrukcia histogramu	24
3.3.2	Úloha histogramu vo verifikácii	25
3.4	Čiarový diagram	27
3.4.1	Konštrukcia čiarového diagramu	27
3.4.2	Úloha čiarového diagramu vo verifikácii	27
3.5	Taylorov diagram	28
3.5.1	Konštrukcia taylorovho diagramu	28
3.5.2	Úloha taylorovho diagramu vo verifikácii	31
4	Návrh vizualizácie	32
4.1	Charakteristika dát	32
4.2	Špecifikácia požiadaviek na vizualizáciu	34
4.2.1	Tri základné požiadavky	34
4.2.2	Užívateľské úlohy	35

4.3	Návrh vizualizácie štatistik verifikácie	37
4.3.1	Prehľad štatistik (<i>Farebná mapa</i>)	37
4.3.2	Detail štatistik (<i>Mnoho-čiarový diagram</i>)	38
4.4	Návrh vizualizácie distribúcie chýb	40
4.4.1	Graf hustoty	40
4.4.2	Pruhový kvantilový diagram	43
4.4.3	Funkčný krabíkový diagram	45
4.4.4	Mapa distribúcií	48
4.5	Návrh farebnej palety	50
4.6	Návrh rozloženia prvkov vizualizácie	51
4.6.1	Viacúrovňový návrh rozloženia prvkov	52
4.6.2	Plochý návrh rozloženia prvkov	53
5	Návrh systému a Implementácia	56
5.1	Návrh systému	56
5.2	Použité technológie	57
5.2.1	Java	58
5.2.2	JavaScript	59
5.3	Verifikačný balík	60
5.3.1	Spracovanie dát	60
5.3.2	Výpočet štatistik	61
5.4	Vizualizačný balík	62
6	Výsledky	64
6.1	Testovanie	64
6.1.1	Testovacia procedúra	64
6.1.2	Testovacie dáta	65
6.1.3	Výsledky testovania	65
6.2	Demonštrácia	68
6.2.1	Štatistiky	68
6.2.2	Chyby	70

6.2.3 Ročný priebeh	72
Záver	73
A Prílohy	75
A.1 GUI	75
A.2 Príklad vizualizácie starého systému	76
A.3 Testovací formulár	77
A.4 CD	78

Zoznam obrázkov

1.1	Vizuálne znázornenie dvoch bežne používaných metód na získavanie hodnôt z mriežky	7
3.1	Porovnanie bodového grafu a Q-Q grafu pre rovnaké dátu. Oba grafy boli vygenerované v programe EVS [NWS15]. a) Bodový graf b) Q-Q graf	21
3.2	Pôvodný návrh krabicového diagramu, ako bol prezentovaný v práci <i>Exploratory Data Analysis</i> (1977) [Tuk77]	23
3.3	Histogram	25
3.4	Porovnanie rôznych dĺžok intervalov. Obrázok je upravený z pôvodného článku [SS07]	26
3.5	Príklad čiarového grafu vygenerovaného v programe Adobe Illustrator	28
3.6	Geometrický vzťah pre popisné štatistiky $R, E', \sigma_r, \sigma_f$	29
3.7	Taylorov diagram [Tay01]	30
4.1	a) Konštrukcia jedného pásu zobrazením hodnôt grafu na farebnú škálu b) Výsledná vizualizácia, ako farebná mapa	38
4.2	a) Bežné zobrazenie viacerých škál b) Nami navrhnuté zobrazenie viacerých škál	39
4.3	Porovnanie dvoch konvenčných techník farbenia so Saitovým dvojtónovým farbením. Obrázok pochádza z pôvodného článku [SMY ⁺ 05].	42
4.4	Porovnanie použitia dvoch techník farbenia. a) Spojité pseudo farbenie b) Dvojtónové pseudo farbenie	43
4.5	Pruhový kvantilový diagram so 4 pruhmi a extrémami. Medián je zvýraznený žltou farbou.	44

4.6	Na obrázku je znázornený príklad pásma $\mathcal{B}(x_1, x_3)$ zloženého z 2 funkcií x_1, x_3 . Taktiež môžme vidieť funkciu x_2 , ktorej graf leží v pásme, zatiaľ čo x_4 podľa BD neleží vôbec a podľa MBD len čiastočne.	47
4.7	Obrázky z článku <i>Functional Boxplots</i> [SG11] a) Funkcie meraní teploty hladiny mora b) Funkčný krabikový diagram c) Rozšírený Funkčný krabikový diagram o centrálne regióny $C_{0.25}$ a $C_{0.75}$	49
4.8	Mapa distribúcií a) Bez farebnej mapy b) Spolu s farebnou mapou	50
4.9	Vývoj farebnej palety pre farebnú mapu	51
4.10	Ikony pre jednotlivé prvky vizualizácie	53
4.11	Schematické zobrazenie viacúrovňového návrhu rozloženia prvkov vizualizácie	54
4.12	Schematické zobrazenie plochého návrhu rozloženia prvkov vizualizácie . .	55
5.1	Schematický popis systému.	57
5.2	Rozdelenie prvkov vizualizačnej pipeline do verifikačného a vizualizačného balíka. Obrázok pipeline pochádza zo stránky http://www.infovis-wiki.net	58
5.3	Triedny UML diagram pre triedy typu DataExtractor.	61
5.4	Triedny UML diagram pre triedy typu ContinuousStatistics.	62
5.5	Príklad štruktúry modulov pre vytvorenie mnoho-čiarového diagramu. . . .	63
6.1	Porovnanie časov nášho (Systém 1) a starého systému (Systém 2)	66
6.2	Výsledky testovania nášho systému pre jednotlivých užívateľov.	67
6.3	a) Farebná mapa zobrazujúca RMSE b) Farebná mapa zobrazujúca MFE .	69
6.4	Zobrazenie RMSE, MFE, MAE a SD (Standard Deviation) na jednom grafe súčasne	69
6.5	Farebná mapa chýb predpovedí pre mesiac Február. Na y -ovej osi je poradie predpovede.	71
6.6	Grafy distribúcie pre mesiac Február a) Graf hustoty b) Pruhový kvantilový diagram c) Funkčný krabikový diagram	71

6.7 Graf ročného priebehu chýb predpovedí. Šedou čiarou sú zobrazené denné štatistiky (konkrétnie RMSE) a červenou je kĺzavý priemer vypočítaný z týchto štatistik.	72
---	----

Zoznam tabuliek

2.1 Porovnanie verifikačného softvéru	18
6.1 Výsledky testovania	65

Zoznam skratiek

WRF	Weather Research and Forecasting
NCEP	National Centers for Environmental Prediction
NCAR	National Center for Atmospheric Research
NWP	Numerical Weather Prediction
GFS	Global Forecast System
NAM	North American Mesoscale Forecast System
RUC	Rapid Update Cycle
SREF	Short Range Ensemble Forecast
GEFS	Global Ensemble Forecast System
ECMWF	European Centre for Medium-Range Weather Forecasts
NOAA	National Oceanic and Atmospheric Administration
ESRL	Earth System Research Laboratory
AFWA	Air Force Weather Agency
NRL	Naval Research Laboratory
CAPS	The Center for Analysis and Prediction of Storms
LES	Large eddy simulation
WMO	World Meteorological Organization
TOGA	Tropical Ocean Global Atmosphere
ME	Mean Error
MFE	Mean Forecast Error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MSE	Mean Square Error
TS	Tracking Signal
MAD	Median Absolute Deviation
GUI	Graphical User Interface
VBA	Visual Basic for Applications
SAS	Statistical Analysis Software
CAWCR	Centre for Australian Weather and Climate Research
IDL	Interactive Data Language
NCL	NCAR Command Language
MET	Model Evaluation Tools
DTC	Developmental Testbed Center
EVS	Ensemble Verification System
HEP	Hydrological Ensemble Prediction
MISE	Mean Integrated Square Error
KDE	Kernel Density Estimation
IMS	Integrated Monitoring System
SVG	Scalable Vector Graphics
CSV	Comma Separated Values
UML	Unified Modeling Language

Úvod

Vizualizácia informácií a vizuálna analýza dát sú dnes veľmi vyvíjaným a moderným odvetvím počítačovej grafiky. Aplikácie vizualizácie informácií sú rôzne od vedeckého výskumu cez finančné analýzy až po komerčné použitie vo verejných médiách. Jej hlavnou úlohou je v prvom rade využitie ľudských kognitívnych schopností pre lepšie porozumenie dát, teda z nezrozumiteľného zhľuku dát vytvoriť zrozumiteľnú vizuálnu reprezentáciu.

Svoje využitie našla vizualizácia informácií aj v procese verifikácie predpovedných modelov počasia. Dnešné metódy verifikácie predpovedí sa sústredia na numerický popis výkonu jednotlivých predpovedných modelov pomocou rôznorodých štatistických metód. Vo výsledku sa teda dáta opisujú ďalšími dátami, avšak práve na vizualizáciu týchto dát sa nekladie príliš veľký dôraz.

V našej práci sme preštudovali používané štatistické metódy pre verifikáciu spojítých premenných, ktoré predpovedá model *WRF*. Hlavným cieľom práce sa však stal návrh a implementácia vizualizačných techník špeciálne navrhnutých pre potreby verifikácie. Sústredili sme sa obzvlášť na to, aby sme ušetrili cenný vizuálny priestor bez straty potrebných informácií. Návrh vizualizácie vychádzal z bežnej meteorologickej praxe, a to zo zaužívaných meteorologických pojmov, štatistických modelov, vizualizačných techník a taktiež úloh, ktoré užívateľ pri verifikácii vykonáva.

Kapitola 1

Verifikácia predpovedných modelov počasia

Verifikácia je proces, ktorý má overiť správnosť fungovania predpovedného modelu počasia. Z tohto dôvodu je nepostrádateľnou súčasťou meteorologického výskumu a taktiež celkového procesu predpovedania počasia [CWS⁺08]. Ciele verifikácie môžeme rozdeliť do troch skupín: *administratívne*, *vedecké* a *ekonomické*. Medzi *administratívne ciele* patrí monitorovanie úspešnosti predpovedania modelu a nasmerovanie užívateľov na jeho správnu konfiguráciu alebo voľbu iného modelu. *Vedeckými cieľmi* sú identifikovanie a oprava slabín modelu a taktiež vylepšovanie predpovedí. *Ekonomickými cieľmi* sú rozhodovanie, kam majú smerovať investície do výskumu a iné závažné ekonomicke rozhodnutia [FJB12].

1.1 Predpovedný model počasia

Už v 19. storočí vývoj termodynamiky na základe Newtonovej fyziky vyvrcholil v ucelení množiny fundamentálnych princípov, ktoré riadia prúdenie plynov v atmosfére. Začiatkom 20. storočia sa o matematický prístup k predpovedaniu počasia najviac zaslúžili osobnosti ako Vilhelm Bjerknes alebo Lewis F.Richardson. Avšak na ďalší úspech, v tejto oblasti, sa muselo čakať až na vynájdenie prvých počítačov počas 2. svetovej vojny ako bol IAS alebo ENIAC [Lyn07]. Prvá úspešná predpoved bola vykonaná v 50. rokoch minulého storočia a to hlavne vďaka práci Jula Charneyho.

Následný vývoj vo výpočtovej sile počítačov, používanie satelitných pozorovaní a vývoj samotnej meteorológie ako vedy zapríčinil, že je numerická predpoved' počasia (NWP) dnes najúspešnejším prístupom ako predpovedať počasie [Gol].

Odtedy vzniklo veľké množstvo modelov, ako sú napríklad GFS, NAM, RUC, WRF, SREF, GEFS, ECMWF, ALADIN a mnoho ďalších. Naša práca sa zameriava konkrétnie na verifikáciu modelu *WRF*, avšak použitie nášho nástroja je univerzálne vďaka spoločnému formátu vystupujúceho z modelov - *Gribu*.

1.1.1 WRF model

Ako sme už spomenuli *The Weather Research and Forecasting* (WRF) model je systém atmosferickej simulácie, ktorým sa realizuje *numerická predpoved' počasia* (NWP).

WRF je podporovaný, ako bežný nástroj pre univerzity, výskum a operatívne komunity, pričom sa usiluje o splnenie všetkých ich požiadaviek súčasne.

WRF model je vhodný pre širokú škálu aplikácií od simulácie jedného búrkového oblaku až po globálne predpovede počasia. Aplikácie modelu WRF vyžadujú numerické predpovede v reálnom čase, vývoj a štúdium asimilácie dát, výskum parametrizovanej fyziky, modelovanie kvality ovzdušia a idealizované simulácie.

V roku 2008 mal model WRF viac ako 6000 užívateľov, no dnes (2015) je evidovaných viac ako 25000 užívateľov vo viac ako 130 krajinách sveta. Tieto fakty poukazujú na to, že WRF model má nie len veľkú základňu užívateľov, ale aj vývojárov¹ a má v budúcnosti istotne svoje miesto a preto si myslíme, že sa oplatí investovať čas a úsilie do verifikácie tohto modelu.

1.2 Dáta

Na správne zhodnotenie úspešnosti modelu potrebujeme dva druhy dát. V prvom rade sa jedná o dáta, ktoré sú výstupom z daného predpovedného modelu počasia, teda **predpovedané dáta**. Tieto umelo získané dáta chceme konfrontovať s realitou, aby sme si mohli

¹Aktívne na vývoji WRF pracuje mnoho ľudí, avšak zasahovať do kódu môže ktokoľvek, keďže je WRF *open source*.

vytvoriť obraz o správnom fungovaní celého modelu. Realitu v našom prípade predstavujú dátá namerané špecializovanými meteorologickými senzormi, ktoré označujeme ako **pozorované dátá** alebo skrátene *pozorovania*.

1.2.1 Predpovedané dátá

Predpovedané dátá z modelu WRF sa ukladajú vo formáte **GRIB**, čo je skratka pre *GRIdded Binary* [WMO94] alebo na iných miestach uvádzané ako *General Regularly-distributed Information in Binary form* [WMO03]. Tento formát je štandardom Svetovej meteorologickej organizácie teda *World Meteorological Organization* (WMO). Jedná sa o pomerne rozšírený formát, používaný pri veľkom množstve meteorologických aplikácií a je taktiež používaný ako výstupný formát pre iné predpovedné modely ako WRF, či už ECMWF, GFS, NAM, SREF alebo mnohé ďalšie [NCE14].

Doteraz boli vyvinuté 3 verzie tohto formátu od 0 po 2. Verzia 0 bola určená pre malé projekty typu TOGA a to iba s limitovaným použitím a dnes sa táto verzia už vôbec nepoužíva. Verzia grib 1 [WMO94] a grib 2 [WMO03] sú dnes bežne používané väčšinou meteorologických centier.

Medzi verziami 1 a 2 nie sú žiadne rozdiely v obsahovej filozofii, preto popis obsahu gribovského formátu, ktorý tu uvádzame je spoločný pre obe tieto verzie. *Gribovský súbor* (ďalej iba *Grib*) pozostáva z viacerých *Gribovských záznamov*, pričom jeden záznam môže existovať ako samostatný Grib. Vďaka tomu je možné ľahko spájať Griby, a to tiež v ľubovoľnom poradí, bez toho, aby sme ich nejako poškodili. Samozrejme musí byť zachovaná homogenita, čo sa týka verzií Gribov, teda verziu 1 nemožno miešať s verziou 2 a naopak. Už samotný názov *Gridded Binary* nám napovedá, že dátá sú usporiadané v pravidelnej mriežke. Každý Gribovský záznam obsahuje dvojrozmernú mriežku (zemepisná šírka x zemepisná dĺžka) hodnôt v určitom čase a vertikálnej hladine. Taktiež v hlavičke záznamu sa nachádzajú *meta-informácie*, ktoré nám hovoria o aké dátá ide, teda o akú premennú sa jedná, čas predpovede, výškovú hladinu a podobne. Grib je zvyčajne z tohto dôvodu 2 až 5 rozmerná dátová štruktúra s veľkým množstvom veličín ako je napríklad teplota, tlak, relatívna vlhkosť, rosný bod, *u* a *v* súradnice vetra a ďalšie, ktoré sú definované v rôznych hladinách. Taktiež je dôležité povedať, že Grib môže zachytávať povrch

celej planéty alebo iba vymedzenú skúmanú oblasť - *doménu*.

1.2.2 Pozorované dátá

Pozorovania sa získavajú meraním priamo v teréne pomocou špecializovaných meracích zariadení, ktoré sú súčasťou meteorologických staníc. Každá stanica môže obsahovať iné vybavenie, napríklad teplomer, zrážkomer, barometer, vetromer a ďalšie podobné [Vas98], ktorými môžeme zachytávať informácie o rôznych skúmaných veličinách.

Majoritná časť meraní sa deje pri povrchu zeme priamo na meteorologických staniciach a nazývajú sa *prízemné* (surface) merania. Tieto merania najlepšie popisujú dianie v oblasti najväčšieho záujmu (biosfére), avšak neobsahujú informáciu o dianí v iných výškových hladinách. Pozorovania týchto hladín sa dejú pomocou *radiosondy*, ktorá je pripojená k meteo balónu alebo vypustená z lietadla smerom k zemi. Takéto pozorovania sa nazývajú *aerologické* (upper air) merania.

Narozenie od predpovedaných dát, pozorované dátá nemajú štandardizovaný formát a zvyčajne sa ukladajú do databázy. Aby sme zhrnuli charakteristiku týchto dát, jedná sa o niekoľko meraných veličín, nameraných v konštantných časových krokoch - napríklad každú minútu alebo každú hodinu - v jednom konkrétnom geografickom bode a zvyčajne pri povrchu zeme, teda ak sa nejedná o aerologické merania, ktoré sa uskutočňujú v štandardných výškových hladinách (V meteorológií či letectve sú to hladiny konštantného tlaku).

1.2.3 Párovanie dát

Z predpovedného modelu a rovnako aj z merania získame veľké množstvo hodnôt. Aby sme mohli korektnie porovnať predpoveď s pozorovami, je nevyhnutné nájsť správne párovanie týchto hodnôt, teda zistiť, ktorú hodnotu porovnať s ktorou, aby sme získali zmysluplný výsledok.

Vždy sa snažíme nájsť správne pozorovanie pre predpoved' a nie naopak. Dôvodom je, že pre jedno pozorovanie existuje viaceré predpovedí a taktiež usporadúvame hodnoty podľa predpovedných hodín a nie chronologicky podľa časovej známky.

Každá hodnota, ktorú chceme spárovať má štyri klúče podľa ktorých hľadáme páry: *veličina* (napríklad teplota), *časová známka*(UTC), *výšková hladina* a *geografická poloha*. Nájsť všetky hodnoty podľa klúča meranej veličiny v Grike je ľahké, keďže sa jedná o kategorickú premennú, teda môže nadobúdať iba určitý konečný počet hodnôt. Toto sa však nedá povedať o čase, hladine a polohe, ktoré sú spojitémi premennými.

Pre čas pozorovania, čas predpovede a výškovú hladinu existujú štandardy, ktoré určujú v akých časoch resp hladinách sa robia merania a predpovede, čo nám uľahčuje prácu. Ak sa napriek tomu daná časová známka nevyskytuje medzi pozorovaniami (napríklad v tom čase vypadlo spojenie alebo sa pokazil prístroj), daný páry sa z párovania vyhadzuje.

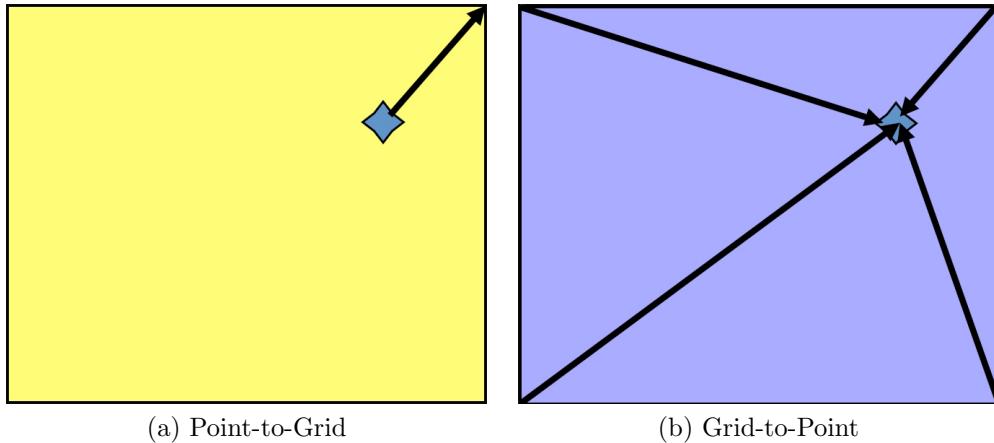
V prípade polohy zo samozrejmých dôvodov neexistuje žiadny štandard a hustota mriežky v Grike nemôže byť nikdy tak veľká, aby poloha našej stanice vždy dopadla na presný bod mriežky. Z tohto dôvodu získavame hodnoty z mriežky z okolitých bodov a to dvoma metódami *Point-to-Grid* a *Grid-to-Point* [FJB12], ktoré sú znázornené na obrázku 1.1. Jedná sa vlastne o dve interpolačné metódy. Point-to-Grid predstavuje metódu *najbližší sused* (*Nearest Neighbour*) a Grid-to-Point *bilineárnu interpolačnú metódu*.

Výber správnej metódy môže značne ovplyvniť výsledok. Dôvodom je, že môžu byť veľké rozdiely hodnôt v okolitých mrežových bodoch (napríklad v členitom teréne) a tak, ak pomocou Point-To-Grid metódy získame nízku hodnotu, tak pomocou Grid-To-Point môžeme získať hodnotu omnoho väčšiu, vplyvom zvyšných troch bodov, ktoré vstúpili do interpolácie. Nemožno však jednoznačne povedať, ktorá z metód je lepšia, keďže obe môžu v istých prípadoch dávať lepšie výsledky.

1.3 Metódy verifikácie

Metódy verifikácie sa delia podľa typu predpovede na 3 druhy [Nur03]:

- Verifikácia kategorickej predpovede
 - Dichotomické (áno/nie) predpovede
 - Mnoho-kategorické predpovede
- Verifikácia pravdepodobnostných predpovedí



Obr. 1.1: Vizuálne znázornenie dvoch bežne používaných metód na získavanie hodnôt z mriežky

- Verifikácia predpovede spojитých premenných

Kategorické predpovede nám rozdeľujú udalosti, v prípade *binárnej predpovede* do dvoch a, v prípade *mnoho-kategorickej predpovede*, do viacerých kategórií. Binárne predpovede alebo inak áno/nie predpovede nám hovoria, či sa daná udalosť udiala alebo nie. Slúžia teda ako upozornenie na javy ako hmla, búrka, blesky, silný vietor a podobne. Na ich verifikáciu slúžia 2×2 kontingenčné tabuľky, z ktorých možno počítať rôzne štatistiky. Mnoho-kategorické predpovede nám rozdeľujú udalosti do viacerých kategórií (zvyčajne 3-4), napríklad v prípade zrážok na: *dážd'/sneženie/mrznúci dážd'*.

Pravdepodobnostné predpovede nám dávajú pravdepodobnosť výskytu nejakej udalosti hodnotami v intervale $\langle 0, 1 \rangle$. Týmto nám umožňujú ľahkú konverziu medzi pravdepodobnostnými a kategorickými predpovedami s použitím *thresholdu*.

V našej práci riešime verifikáciu spojitých premenných, ktorá zvyčajne využíva štatistické metódy určujúce ako veľmi sa daná predpoved lísi od pozorovania. Tieto metódy opisujeme v ďalšej sekcií tejto kapitoly.

1.4 Meranie chyby predpovede

Výsledkom procesu párovania je n párov (predpoved', pozorovanie), ktoré je možné porovaňať. Z porovnania týchto dvojíc získame numerickú hodnotu, ktorá nám hovorí o veľkosti chyby predpovede daného modelu pre vybrané predpovedané časy.

Chybu predpovede e_i pre i -tu dvojicu (y_i, \hat{y}_i) definujeme takto:

$$e_i = (y_i - \hat{y}_i)$$

Kde y_i je predpoved' a \hat{y}_i je pozorovanie. Takýmto spôsobom z n párov získame n chýb, ktoré agregujeme pomocou rôznych štatistických metód, ktoré sú bežne používané pri verifikácii predpovedí, ako sa spomína v [Nur03], [FJB12] a [Cas09]. Výsledkom agregácie je numerická hodnota, ktorá sa nazýva *skóre* predpovede.

1.4.1 Stredná chyba predpovede

Budeme ju označovať ako *MFE* z anglického *Mean Forecast Error*, ale v literatúre je možné ju nájsť ako *ME* [Nur03], teda *stredná chyba* alebo ako *Linear Bias* [Cas09, FJB12]. Vzorec pre výpočet MFE vyzerá nasledovne:

$$MFE = \frac{1}{n} \sum_{i=0}^n e_i$$

MFE je možné vypočítať aj ako rozdiel priemerov predpovedí a pozorovaní.

$$MFE = \bar{y} - \bar{\hat{y}}$$

MFE vyjadruje priemerný smer chyby. To znamená, že pozitívny výsledok indikuje *over-forecast*, teda nadhodnotenú predpoved' a negatívny výsledok *under-forecast*, teda podhodnotenú predpoved'. Avšak MFE **nevyjadruje veľkosť** chyby v tomto smere, keďže kladné a záporné chyby sa navzájom môžu zrušiť. Napríklad máme množinu chýb $E = \{2, -5\}$, tak MFE pre E je -1.5, ale priemerná veľkosť chyby je 3.5.

1.4.2 Stredná absolútна chyba

Budeme ju označovať ako *MAE* z anglického *Mean Absolute Error*. Vzorec pre výpočet MAE vyzerá nasledovne:

$$MAE = \frac{1}{n} \sum_{i=0}^n |e_i|$$

Narozdiel od MFE **neurčuje smer chyby**, ale vyjadruje veľkosť chyby. Z týchto dôvodov je v praxi odporúčané zobrazovať MFE a MAE súčasne [Nur03].

1.4.3 Stredná kvadratická chyba

Budeme ju označovať ako *RMSE* z anglického *Root Mean Square Error*. Vzorec pre výpočet RMSE vyzerá nasledovne:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^n e_i^2}$$

Z povahy vzorca pre RMSE je jasné, že rovnako ako MAE, ani RMSE neurčuje smer chyby, pretože nadobúda vždy iba kladné hodnoty. Ďalšou vlastnosťou RMSE je, že nadobúda hodnoty vždy väčšie alebo rovné ako MAE, pričom výsledok RMSE je citlivý na veľké hodnoty chýb.

V praxy sa zvykne používať aj *MSE* (*Mean Square Error*):

$$MSE = \frac{1}{n} \sum_{i=0}^n e_i^2$$

Má podobné vlastnosti ako RMSE s jediným rozdielom, že RMSE meria veľkosť chyby zachovávajúc jednotky danej veličiny (napr. °C), zatiaľ čo MSE jednotky nezachováva [Nur03]. Preto sme si pre nás účel zvolili RMSE, ktoré je jednoduchšie zobraziť spolu s MFE a MAE v jednom grafe, keďže sa zachováva konzistentnosť jednotiek veličín.

1.4.4 Všeobecná kumulovaná chyba

V našom systéme sme navrhli všeobecný vzorec na výpočet kumulovaného skóre, ktorým možno vyjadriť ľubovoľnú zo spomenutých štatistických metód. Takéto vyjadrenie umožňuje

nie len všeobecnosť, ale aj jednoduché rozšírenie systému o ďalšie metódy a to nie len programátorom, ale aj samotným užívateľom systému.

Všeobecný vzorec na výpočet *skóre* pre danú predpoved' vyzerá takto:

$$Score = \Phi\left(\sum_{i=0}^n \varepsilon(e_i)\right)$$

Kde Φ je ľubovoľná funkcia z \mathbb{R} do \mathbb{R} , teda $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ a podobne funkcia $\varepsilon : \mathbb{R} \rightarrow \mathbb{R}$. Spomenuté metódy môžme teda skonštruovať za definovaním správneho Φ a ε .

Napríklad pre *MFE*:

$$\Phi(x) = \frac{x}{n}$$

$$\varepsilon(e) = e$$

Pre *MAE*:

$$\Phi(x) = \frac{x}{n}$$

$$\varepsilon(e) = |e|$$

Pre *RMSE*:

$$\Phi(x) = \sqrt{\frac{x}{n}}$$

$$\varepsilon(e) = e^2$$

Pre *MSE*:

$$\Phi(x) = \frac{x}{n}$$

$$\varepsilon(e) = e^2$$

Ako sme spomenuli, je možné rozšírenie o ďalšie metódy a to napríklad o Brownov a Triggov *signál chybných predikcií*, ktorý budeme označovať ako *TS* z anglického *Tracking Signal*. Tieto metódy sme vyššie nespomenuli, keďže sa v meteorologickej praxi nepoužívajú. Uvádzame ich však ako možné rozšírenie, keďže sú tieto metódy bežne používané pri verifikácii iných predpovedných modelov, ako sú tie meteorologické.

1.4.5 Medián absolútnych chýb

Budeme ju označovať ako MAD z anglického *Median Absolute Deviation*. Vzorec pre výpočet MAD vyzerá nasledovne:

$$MAD = \text{median}(|e|) = |\tilde{e}|$$

Nech je daná usporiadaná postupnosť Y_1, \dots, Y_N , tak potom *median* náhodnej premennej x je definovaný rovnako ako v [Wei14]:

$$\text{median}(x) = \tilde{x} \equiv \begin{cases} Y_{(N+1)/2} & \text{ak } N \bmod 2 = 0 \\ \frac{1}{2}(Y_{(N+1)/2} + Y_{(N+1)/2+1}) & \text{ak } N \bmod 2 = 1 \end{cases}$$

Z daného vzorca môžeme vidieť podobné vlastnosti ako má MAE, avšak MAD je robustnejší a extrémne chyby nemajú na skóre žiadnen efekt.

Kapitola 2

Predchádzajúce riešenia 1

Verifikačný softvér

Verifikácia predpovedných modelov počasia je úloha dokonale stvorená pre automatizáciu. Z tohto dôvodu meteorológovia začali využívať dostupný štatistický softvér a neskôr boli taktiež vyvíjané špecializované nástroje určené pre verifikáciu. Môžeme teda rozdeliť verifikačný softvér do dvoch základných kategórií a to *statistický* a *špecializovaný*, ktorý je zväčša podporovaný rôznymi národnými a medzinárodnými organizáciami.

2.1 Štatistický softvér

Spoločnými črtami štatistických programov, ktoré v tejto časti spomenieme sú:

- Obmedzená verifikačná funkčnosť
- Slabé / Žiadne GUI
- Potrebná znalosť špecifického programovacieho jazyka

Taktiež ide zväčša o platený softvér s pomerne vysokými cenami licencií. No aj napriek týmto slabým stránkam sú často používané v meteorologickej komunite, či už kvôli dobrej podpore, veľkému množstvu návodov, alebo kvôli tomu, že poskytujú rôzne štatistické funkcie a umožňujú rýchle doimplementovanie nových verifikačných metód.

2.1.1 Tabuľkový softvér

Napriek tomu, že je tabuľkový softvér na výpočet štatistik zamietnutý komunitou vedcov a štatistikov ako nevhodný a neprofesionálny, tak je využívaný, a to pomerne často, aj vo vedeckých kruhoch. Výhodou je, že novému užívateľovi umožňuje okamžite vidieť všetky kroky v základných procedúrach verifikácie a teda je výborný pre výučbové účely [Poc11]. Najznámejší kus softvéru z pomedzi kommerčných produktov je *Microsoft Excel* [Mic15] a z voľne dostupných je jeho opensource náprotivok *Open Office Calculate* [Ope15]. Oba programy zahrňujú základné štatistické funkcie ako napríklad stredná kvadratická chyba (*MSE*) pre spojité predpovede (pozri odsek 1.4.3) a taktiež umožňujú generovanie jednoduchých grafov na základe tabuľkových dát. Tabuľkový softvér neposkytuje priamo funkciu na výpočet ďalších sofistikovanejších verifikačných štatistik, avšak umožňuje ich implementáciu pomocou makro programovania v špecifickom jazyku. Pre Microsoft Excel je to *Microsoft Visual Basic for Applications* (VBA) [Mic13] a pre Open Office Calculate zasa *OpenOffice.org Basic* [Ope13]. Oba jazyky patria do rodiny *Basic* jazykov, takže majú mnoho podobných črt.

2.1.2 MATLAB

MATLAB je interaktívne prostredie s vlastným programovacím jazykom, ktorý je využívaný miliónmi inžinierov a vedcov po celom svete [TM15] a tým nevynímajúc meteorológov a ďalších odborníkov pracujúcich v atmosférickom výskume. Zvyčajne sa MATLAB využíva na výskum a prototypovanie nových metód a procedúr [Poc11], pretože umožňuje rýchlu a jednoduchú implementáciu, keďže jeho súčasťou je mnoho matematických knižníc a je prispôsobený na prácu s maticami dát. Výhodou MATLABu je, že umožňuje tvorbu GUI a taktiež poskytuje kreslenie rôznorodých grafov a diagramov. Mali by sme však podotknúť, že podobne ako väčšina štatistického softvéru, aj *MATLAB* je kommerčný produkt. Jeho cena za jednu licenciu je \$2,650 (k roku 2015), čo je pomerne vysoká suma, ak vezmeme do úvahy za akým účelom chceme tento softvér využívať a ako dobre je naň prispôsobený.

2.1.3 R

Často používaným a pomerne mocným nástrojom je *open source* skriptovací jazyk *R* [Fou15]. V posledných desaťročiach sa stal dominantným jazykom v oblasti štatistického výskumu. Napriek tomu, že ide o voľne stiahnutelný softvér, tak jeho základný balík obsahuje všetky funkcie, ktoré obsahujú aj platené produkty. R-ko však nezostáva len pri tom, pretože v dobe písania tejto práce (marec 2015) bolo dostupných vyše 6400 užívateľských balíkov s rôznorodou funkcionalitou. Pre nás je dôležité spomenúť, že medzi týmito balíkmi sa objavil aj balík určený na verifikáciu s názvom **verification** [Lab14]. Tento balík obsahuje základné funkcie verifikácie na výpočet štatistik pre spojité, kategorické ale i pravdepodobnostné predpovede.

Jazyk R neslúži iba na rôznorodé štatistické výpočty, ale poskytuje aj veľmi dobre parametrizovateľnú vizualizáciu. V balíkoch jazyka sa nachádzajú funkcie pre čiarové diagramy, krabicové diagramy, bodové grafy a mnohé iné komplexnejšie vizualizácie, ale tiež funkcie na zobrazenie základných vizuálnych prvkov, ktorými možno vytvoriť úplne novú osobitnú vizualizáciu.

2.1.4 Statistical Analysis Software (SAS)

Statistical Analysis Software [Ins15], skrátene SAS, je opäť štatistický programovací jazyk aj so svojim vývojovým prostredím. V oblasti bioštatistiky a farmakológie je veľmi uznávaným a často používaným jazykom. Ked'že je veľká podobnosť v používaných metódach medzi verifikáciou predpovedí a spomínanými odvetviami [Poc11], SAS poskytuje funkciu použiteľnú aj pre verifikáciu. Okrem iného SAS ponúka základné, ale aj niektoré pokročilejšie nástroje na vizualizáciu dát. Opäť však musíme podotknúť, že ide o komerčný produkt, ktorého cena licencie je pomerne vysoká.

2.1.5 Interactive Data Language (IDL)

IDL, teda *Interactive Data Language* [Sol15] je opäť jeden z matematických programovacích jazykov, ktoré patria medzi menej používané v komunite atmosferického výskumu [Poc11]. Napriek tomu niektorí výskumníci medzi ktorými je aj *Beth Ebert* z *Centre for*

Australian Weather and Climate Research (CAWCR) uverejnili na svojich webstránkach kód obsahujúci metódy verifikácie napísané v IDL:

- Metódy pre verifikáciu pravdepodobnosti zrážok (<http://www.cawcr.gov.au/projects/verification/POP3/POP3.html>)
- Priestorové metódy (<http://www.cawcr.gov.au/staff/eee/#Interests>)

IDL na používanie požaduje taktiež získanie platenej licencie, čo obmedzuje počet užívateľov a rovnako aj zdieľanie kódu, ktorý by si mohol ktokoľvek spustiť.

2.2 Špecializovaný softvér

Tento typ softvéru je často podporovaný veľkými meteorologickými inštitúciami, ktoré majú veľa skúseností v tejto oblasti. Preto obsahuje zvyčajne špecializované metódy na prácu s meteorologickými dátami alebo aj priamo metódy slúžiace pri verifikácii predpovedných modelov počasia. Jedná sa výlučne o open source produkty, ktorých vývoj je dobre finančovaný a neustále napreduje. Ďalšou spoločnou črtou je, že všetky spomenuté programy poskytujú viacmenej rovnaké vizualizačné nástroje, v ktorých vidíme priestor na výrazné vylepšenia.

2.2.1 NCAR Command Language (NCL)

Ako väčšina softvérových riešení v predchádzajúcej sekcii, tak aj *NCL*, teda *NCAR Command Language* [UCA15a] je skriptovací jazyk s vlastnou syntaxou a interpreterom. Ide o voľne šíritelný produkt od *National Center for Atmospheric Research* (NCAR), a jeho zameranie je pochopiteľne na atmosferický výskum. Jeho primárnym cieľom je spravovanie a manipulácia klimatologických dát z predpovedných modelov, čo je jedna zo základných častí verifikácie. NCL obsahuje balík funkcií, ktorými je ľahké implementovať štatistiky verifikácie spojítých premenných, ktoré sme spomínali v časti 1.4.

Súčasťou jazyka je aj možnosť vizualizácie predpovedných dát, ale taktiež aj niektoré jednoduché štatistické vizualizačné nástroje použiteľné vo verifikácii. Vzhľadom na

to, že ide o meteorologický softvér, tak množstvo funkcií slúžiacich na verifikáciu je nedostačujúci, čoho príčinou môže byť, že verifikácia predpovedných modelov je ešte stále vo svojich začiatkoch [Poc11]. Sme si istí, že ako metódy aj používané praktiky pokročia, tak budú komunitou NCL vytvorené funkcie slúžiace na tento účel.

2.2.2 Model Evaluation Tools (MET)

Jeden z najprepracovanejších softvérov z oblasti verifikácie predpovedných modelov je bezpochyby *MET*, teda *Model Evaluation Tools* [DTC15]. Jeho autorstvo je pripísané opäť americkej organizácii *NCAR* v spolupráci s *Developmental Testbed Center* (DTC) a celý projekt financovala agentúra *AFWA* spolu s *NOAA*.

Rovnako ako naša aplikácia, aj MET sa sústredí na verifikáciu modelu WRF a taktiež môže byť rozšírený na použitie pre iné predpovedné modely počasia alebo iné typy predpovedí. O tomto svedčia aj niektoré z hlavných filozofických cieľov pri návrhu aplikácie a tými sú *modularita* a *prispôsobivosť* softvéru. Samotný nástroj MET nie je teda samostatne existujúcou aplikáciou, ale skladá sa z viacerých modulov, ktoré môžu fungovať aj ako samostatné nástroje a taktiež môže byť MET týmto spôsobom ľahko rozšíriteľný [Cen14].

Čo sa týka verifikácie, tak MET sa sústredí na verifikáciu spojitych premenných a to na bodovo ale aj objektovo založenými verifikačnými metódami. *Bodovo založené* sú orientované na verifikáciu v konkrétnom geografickom bode, zatiaľ čo *objektovo založené* pristupujú k dátam geometricky a identifikujú objekty, ktorými môžu byť oblasti s istou tlakovou hladinou, búrky, oblačnosť a podobne.

Na vizualizáciu slúžia *Grid-Stat tool*, ktorý umožňuje výpočet štatistik a ich vizualizáciu pre bodovo založené metódy, zatiaľ čo *MODE tool* poskytuje túto funkcionality pre objektovo založené metódy [Cen14]. Oba nástroje neposkytujú nijak zvlášť veľkú vizualizačnú silu, avšak v roku 2011 vznikol produkt *METViewer* [OGJ11], ktorý sa snaží tento problém riešiť. METViewer je webová aplikácia, špeciálne navrhnutá pre výstupy z MET, ktoré spracúva a vizualizuje pomocou R skriptov. Samotná aplikácia je vo verzii 1.0 ako aj 1.1 dostupná na webe (<http://www.dtcenter.org/met/metviewer/metviewer.jsp>, http://www.dtcenter.org/met/metviewer/db/mv_hmt_2010) a podporuje všetky typy diagra-

mov, ako väčšina spomenutého softvéru, teda krabicové, bodové a čiarové diagramy a taktiež histogramy.

Silnými stránkami MET teda zostáva silná podpora inštitúcií, ktoré podporujú a finan- cujú vývoj. Vďaka tomu sa vývoj posúva stále dopredu, organizujú sa rôzne workshopy a je dostupné množstvo návodov. Nevýhodou je silná závislosť na softvérových riešeniach tretej strany a taktiež silná modularita, čo zapríčinuje komplikovanú inštaláciu softvéru.

2.2.3 Ensemble Verification System (EVS)

EVS, teda *Ensemble Verification System* [NWS15] je program vyvíjaný pod záštitou sku- piny s názvom *Hydrological Ensemble Prediction* (HEP), ktorá patrí pod oddelenie *Office of Hydrologic Development (OHD)*, patriace do *National Weather Service* [Bro15].

Narozdiel od MET, EVS nepodporuje získavanie dát z rôznych modelových dátových formátov ako je GRIB1, GRIB2, BUFR a podobne (pozri sekciu 1.2), ale je potrebné, aby boli pozorovania aj predpovede vo formáte určenom pre EVS. Jeden súbor predstavuje ta- buľku hodnôt oddelených medzerami, kde prvá hodnota v riadku je časová známka. EVS podporuje rôzne štatistiky pre predpovede spojitých premenných, ale taktiež aj pravde- podobnostné predpovede, ktorých výpočet je podrobne popísaný priamo v programe pri konfigurácii verifikácie.

Výsledné štatistiky, môžme v EVS vizualizovať pomocou už mnohokrát spomínaných diagramov, ktorých podrobný popis nájdeme v nasledujúcej kapitole. Výstupy z EVS je potom možné uložiť do formátu *png*, alebo ako mnoho-stránkový *pdf* súbor.

V závere môžme povedať, že EVS je kvalitný softvér ktorý sa sústredí na verifikáciu hydrologických a hydrometeorologických premenných. Na jeho ďalšom vývoji sa neustále pracuje a v dobe písania tejto práce (20.1. 2015) bola zverejnená verzia 5.4 [NWS15].

2.3 Zhrnutie

V tejto časti sme sa snažili prehľadným spôsobom v tabuľke 2.1 zhrnúť porovnanie spo- menutého softvéru využívaného pri verifikácii.

Tabuľka 2.1: Porovnanie verifikačného softvéru

Názov	Štatistický / Špecializovaný	Programovací jazyk	Open Source	Zameranie	Verifikácia	Vizualizácia
Tabuľkový softvér	Štatistický	NIE	Rôzne	Rôzne	Žiadne explicitné verifikačné funkcie. Možnosť doimplementovať.	Základné diagramy
MATLAB	Štatistický	ÁNO	NIE	Štatistický výskum, Akademické účely	Žiadne explicitné verifikačné funkcie. Možnosť doimplementovať.	Vizualizačné nástroje v rámci jazyka
R	Štatistický	ÁNO	ÁNO	Štatistický výskum	Mnoho štatistických funkcií. Verifikačný balík 'verification'.	Veľmi dobré parametrizovateľná vizualizácia. Diagramy ľubovoľného typu.
SAS	Štatistický	ÁNO	NIE	Bioštatistika, zdravotníctvo	Funkcionalita príbuzná verifikačnej	Vizualizačné nástroje v rámci jazyka
IDL	Štatistický	ÁNO	NIE	Atmosferické vedy	Niektoré užívateľmi vytvorené knižnice.	Vizualizačné nástroje v rámci jazyka
NCL	Špecializovaný	ÁNO	ÁNO	Predpovedanie počasia a výskum v tejto oblasti	Niekoko verifikačných funkcií.	Vizualizačné nástroje v rámci jazyka
MET	Špecializovaný	NIE	ÁNO	Predpovedanie počasia a výskum v tejto oblasti	Navrhnuté pre verifikáciu.	Nástroj METViewer. Základné diagramy.
EVS	Špecializovaný	NIE	ÁNO	Predpovedanie počasia a výskum v tejto oblasti	Navrhnuté pre verifikáciu.	Základné diagramy

Kapitola 3

Predchádzajúce riešenia 2

Techniky vizualizácie vo verifikácii

3.1 Bodový graf

Najjednoduchším spôsobom ako analyzovať vzťah dvoch náhodných premenných je *bodový graf*, ktorý je inak nazývaný aj *korelačný diagram* a známy je tiež pod svojim anglickým pomenovaním *scatter plot*. Bodový graf je vhodný na štúdium kolerácie dvoch premenných a taktiež je výborný pri odhalovaní takzvaných *outlier-ov*, teda hodnôt, ktoré sa nejakým spôsobom výrazne odlišujú od tých ostatných. Taktiež nám nepriamo podáva správu o distribúcii hodnôt, čo však pri ich veľkom počte môže byť skreslené, keďže sa body začnú postupne prekrývať, a tak nemožno určiť v akej oblasti je viacej, či menej bodov. Tento jav sa nazýva *overplotting*.

3.1.1 Konštrukcia bodového grafu

Skonštruovanie bodového grafu je veľmi jednoduché a aj preto je často používaným prostriedkom na vizualizáciu. Na svoju konštrukciu využíva kartesiánsku sústavu súradníc. Dve náhodné premenné, ktoré chceme porovnať, vizualizujeme tak, že spravíme zobrazenie jednej premennej na x -ovú súradnicovú os a zobrazenie druhej premennej na y -ovú os. Následne v danom bode nakreslíme určený symbol, čo zvyčajne býva čierna bodka alebo krúžok. Na obrázku 3.1a) vidíme príklad výsledného bodového grafu, ktorý vznikne

takýmto postupom.

Bodový graf ponúka mnoho spôsobov, ako pridať ďalší rozmer informácie do vizualizácie. Môžeme zvoliť odlišnú súradnicovú sústavu, čím veľmi jednoducho vytvoríme napríklad 3D bodový graf pre trojrozmerné dátu. Ďalšou možnosťou rozšírenia je zmena vykreslovaného symbolu, ktorému môžeme nastavovať rôzne parametre, do ktorých možno zakódovať nové informácie. Zvyčajne sú týmito parametrami farba, alfa-transparencia [Few08], veľkosť [VWvH⁺07], tvar [CCM13, JHM⁺13] a podobne. Tieto modifikácie si v komunite vizuálnej analýzy vyslúžili aj vlastné názvy a teoretické zázemie, avšak v našej práci sa týmto druhom diagramov nebudeme venovať.

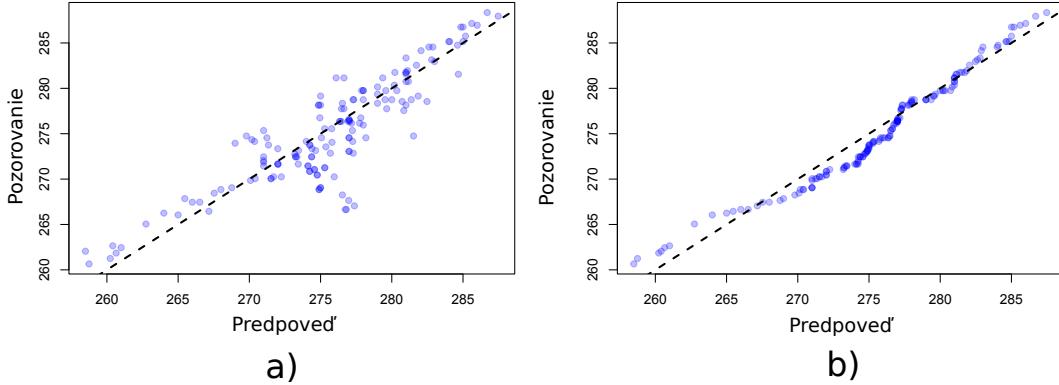
3.1.2 Kantil-kvantil graf

Dôležitou a často využívanou variáciou pre bodový graf je *kvantil-kvantil graf*, skrátene *Q-Q graf* (Q z anglického *quantil*). Jediným rozdielom medzi bodovým grafom a Q-Q grafom je, že zatiaľ, čo bodový graf vizualizuje surové dátu, Q-Q graf zobrazuje iba kvantily z oboch dátových množín.

Kvantilmi sú hodnoty, ktoré rozdeľujú usporiadanú dátovú množinu na niekoľko rovnako veľkých častí. Pre jednu dátovú množinu je $(q - 1)$ q -kvantilov. Najznámejším príkladom je 2-kvantil, ktorý sa nazýva medián. Ďalšími bežnejšie používanými sú 4-kvantily, 10-kvantily a 100-kvantily, čo sú vlastne kvartily, decily a percentily.

To, čo sa zvyčajne robí pri Q-Q grafe je, že sa nevyberá konkrétny typ kvantilu, ale hodnoty sa jednoducho usporiadajú podľa veľkosti a zobrazia. Ak máme množinu A a jej veľkosť je $n = |A|$, tak vizualizjeme n q -kvantilov, kde $q = n + 1$. Tým, že nezobrazujeme na grafe surové dátu, ale kvantily, k -ty bod v grafe nie je zobrazením k -teho páru hodnôt, ako to bolo v klasickom bodovom grafe, ale zobrazením k -teho q -kvantilu.

Na obrázku 3.1 môžme vidieť porovnanie bodového grafu a Q-Q grafu pre rovnakú dátovú množinu.



Obr. 3.1: Porovnanie bodového grafu a Q-Q grafu pre rovnaké dátá. Oba grafy boli vygenerované v programe EVS [NWS15]. a) Bodový graf b) Q-Q graf

3.1.3 Úloha bodového grafu vo verifikácii

Úloha bodového grafu a Q-Q grafu vo verifikácii je rovnaká, avšak oba nám dávajú trocha iný pohľad na dátá. Vo všeobecnosti nám tieto dva grafy dávajú informáciu o vzťahu medzi dvoma veličinami, ich koreláciu a taktiež aj o ich distribúcii.

Dôležitým faktorom teda zostáva aké premenné umiestnime na x -ovú a aké na y -ovú os. Vo verifikácii predpovedných modelov počasia sú to zvyčajne predpovede na jednej osi a pozorovania na druhej. Taktiež sa zvyknú robiť dvojice (predpoveď, chyba predpovede), (pozorovanie, chyba predpovede) alebo (čas predpovede, chyba predpovede) a im podobné. V našej aplikácii sme testovali poslednú zo spomenutých dvojíc.

3.2 Krabiový diagram

Krabiový diagram je v anglickej literatúre zvyčajne nazývaný *box plot* alebo na niektorých miestach označovaný tiež ako *box and whisker*¹ *plot*. Odkedy bol prvýkrát publikovaný v roku 1977 [Tuk77], uplynulo už takmer 40 rokov a dnes ho považujeme už za štandardnú techniku ako vizualizovať distribúciu hodnôt kompaktným spôsobom. Na svoju reprezentáciu využíva súbor 5 čísel (tzv. *5-number summary*) [Pot06], ktoré charakterizujú distribúciu dát robustným spôsobom. Tým, že zredukujeme zvyčajne veľkú dátovú množinu na týchto päť hodnôt ušetríme nielen vzácný vizuálny priestor [WS12], ale taktiež námahu

¹Slovo *whisker* znamená po slovensky fúz, čo naznačuje, že ciary, ktoré spájajú horný a dolný kvartil s hraničnými hodnotami pripomínajú fúzy.

analytika, ktorý sa snaží preskúmať iba niektoré vybrané charakteristiky.

3.2.1 Konštrukcia krabicového diagramu

Na zostavenie krabicového diagramu potrebujeme týchto 5 hodnôt: medián, horný a dolný kvartil, maximum, minimum (pozri obrázok 3.2). Prvé tri hodnoty sú takzvané kvartily (Q_1 , Q_2 , Q_3), ktoré rozdeľujú súbor dát na 4 rovnako veľké časti a ďalšie dve sú extrémne hodnoty, ktoré ohraničujú celú dátovú množinu (pozri podsekcii 3.1.2 pre vysvetlenie pojmu kvantil).

Kvartil Q_2 je medián hodnôt a je definovaný rovnako ako v časti 1.4.5. Ďalej horný (Q_1) a dolný (Q_3) kvartil získame ako medián hodnôt pod a nad hodnotou Q_2 , pričom hodnotu Q_2 nezahŕňame do výpočtov.

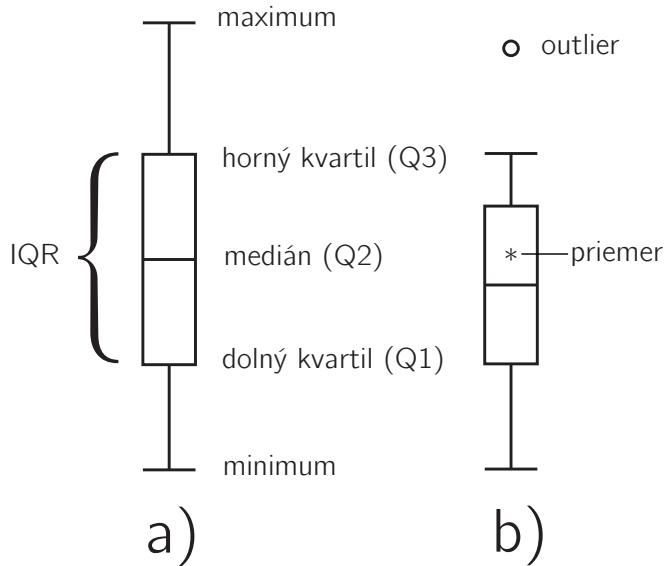
Na obrázku 3.2 vidíme, že krabica v grafe určuje pozície horného a dolného kvartilu, zatiaľ čo vnútro krabice znázorňuje takzvané *IQR*. Táto skratka označuje *interquartile range*, čo sa dá preložiť ako *medzikvartilový rozsah*. IQR definujeme ako rozdiel kvartilov Q_3 a Q_1 :

$$IQR = Q_3 - Q_1$$

IQR nám hovorí o vzdialosti týchto dvoch kvartilov, preto nám môže byť tento vzorec na pohľad podozrivý, keďže sa javí, že IQR by mohlo nadobúdať aj záporné hodnoty. My však vieme z definície Q_3 a Q_1 , že $Q_3 > Q_1$ a ich rozdiel je teda vždy nezáporný (Hovoríme o rozdieli Q_3 od Q_1 , tak ako je definované IQR).

Malú obmenu pôvodného návrhu krabicového diagramu od Tukeyho, vidíme na obrázku 3.2 b), kde malé bodky znázorňujú hodnoty nazývané *outlier*, teda hodnoty ležiace ďaleko od hlavného dátového tela, a hviezdička v strede diagramu určuje priemer hodnôt. Môžeme si všimnúť, že konce čiar vychádzajúcich z boxu nemôžu byť extrémy celej množiny dát, ale sú iba extrémami vypočítaných z dát bez *outlier*-ov.

Otázkou zostáva ako určiť, ktorá hodnota je *outlier* a ktorá nie je. Na zodpovedanie tejto otázky sa využíva už spomínaný rozsah IQR. Pomocou neho sa definujú hranice *inner fences* (f_1, f_2) a *outer fences* (F_1, F_2), za ktorými hovoríme už o *outlier*-och alebo o ďalekých *outlier*-och [SOA04]. Definované sú nasledovne:



Obr. 3.2: Pôvodný návrh krabicového diagramu, ako bol prezentovaný v práci *Exploratory Data Analysis*(1977) [Tuk77]

$$f_1 = Q1 - c \times IQR$$

$$f_2 = Q3 + c \times IQR$$

$$F_1 = Q1 - C \times IQR$$

$$F_2 = Q3 + C \times IQR$$

Konštanty c a C sú v niektorých zdrojoch definované rôzne. Najčastejšie sa však vyskytujú hodnoty $c = 1.5$ a $C = 3$, tak ako ich určil pôvodný autor krabicového diagramu [Tuk77].

3.2.2 Úloha krabicového diagramu vo verifikácii

Ako sme spomenuli v úvode tejto sekcie, vo všeobecnosti je úlohou krabicového diagramu zobraziť distribúciu dát v kompaktnom tvaru a teda slúži na rýchle porovnanie distribúcií viacerých skupín dát. Pri verifikácii predpovede spojitéhých premenných sa používa na viačero účelov a my tu spomenieme len niekoľko z nich.

V prvom rade ide o porovnanie distribúcie predpovedí s distribúciou pozorovaní za istý časový interval. V takomto prípade máme vedľa seba iba dva krabicové diagramy, ktoré

navzájom porovnávame. Použitie krabicového diagramu v takomto prípade, kedy jeden graf pozostáva iba z niekoľkých (2 až 4) krabicových diagramov považujeme za zbytočné. Pri takomto počte nie je potreba na redukciu vizuálnych prvkov a existujú lepšie techniky na vizualizáciu distribúcie, ktoré sprostredkúvajú viacej informácie a teda môže byť analýza efektívnejšia.

Ďalším použitím je porovnanie distribúcie chýb predpovedí, či už pre rôzne meraenia, konkrétnie predpovedané časy, rôzne predpovedné modely a podobne. Tu považujeme použitie krabicového diagramu za opodstatnené, keďže ide zväčša o porovnávanie väčšieho množstva distribúcií, a tak je potrebná jeho jednoduchosť, čitateľnosť, kompaktnosť a iné jeho vlastnosti.

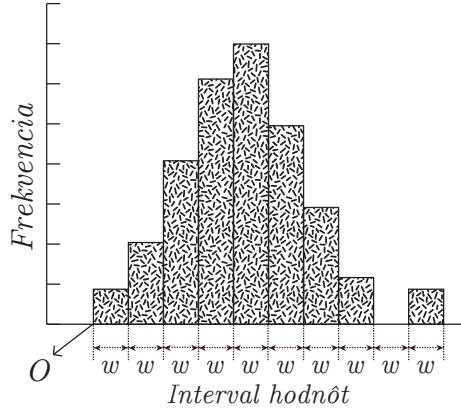
3.3 Histogram

Ďalším veľmi bežným spôsobom, ako vizualizovať distribúciu dát je pomocou *histogramu*, ktorého existencia sa datuje do roku 1895, kedy ho uviedol vo svojej práci Karl Pearson [Pea95]. Histogram využíva veľmi jednoduchú myšlienku vizualizovať frekvencie hodnôt ako stĺpce rôznej výšky, ktorá sa v priebehu histórie analýzy dát ukázala ako veľmi užitočná.

3.3.1 Konštrukcia histogramu

Konštrukcia histogramu je pomerne jednoduchá. V prvom rade potrebujeme rozdeliť interval hodnôt na disjunktné podintervaly rovnakej dĺžky. Na vygenerovanie týchto intervalov využíva autor diagramu počiatok O a veľkosť intervalu w . Prvý interval je $(O, O + w)$ a všetky nasledujúce vzniknú posúvaním intervalu o dĺžku w v kladnom smere. Ďalej pre každý takto vzniknutý interval vypočítame kolko hodnôt doň spadá. Potom už môžme vykresliť jednotlivé stĺpce pre každý interval ako obdlížniky so šírkou w a výškou, ktorá sa určí na základe početnosti hodnôt pre daný interval. Na obrázku 3.3 môžeme vidieť takýmto spôsobom vygenerovaný histogram.

Z vyššie popísaného postupu a rovnako aj z obrázka 3.4 môžme vidieť, že konštrukcia histogramu závisí od zvoleného počiatku O , no v prvom rade od zvolenej veľkosti intervalu



Obr. 3.3: Histogram

w . Z tohto dôvodu vzniklo mnoho prístupov, ktoré sa snažia optimalizovať w , tak aby sa získal optimálny histogram. Optimálny histogram $h(x)$ je taký, ktorý minimalizuje integrovanú strednú kvadratickú chybu ($MISE$) vzhľadom na pôvodnú funkciu hustoty $f(x)$. Túto chybu vypočítame takto: $MISE = \int (h(x) - f(x))^2 dx$.

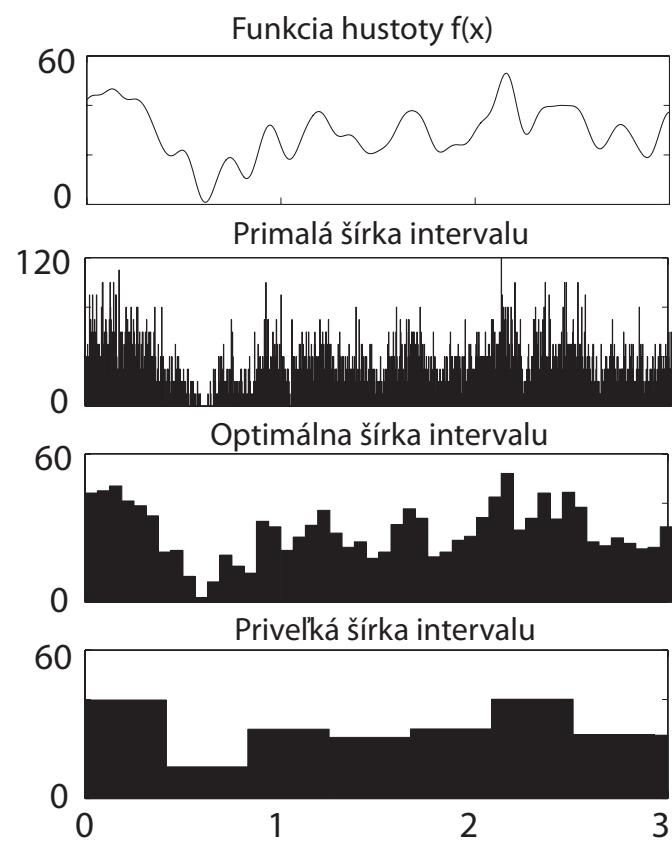
Jedným z príkladov, ktorý toto využíva je algoritmus na generovanie optimálnej hodnoty w , ktorý navrhol *Shimazaki* a *Shinomoto* [SS07]. Autori algoritmu požívajú dekompozíciu MISE na vytvorenie cenovej funkcie $C_n(w)$ pre dĺžku intervalu w :

$$C_n(w) = \frac{2\bar{k} - v}{(nw)^2}$$

kde n je počet všetkých hodnôt množiny, \bar{k} je priemerný počet hodnôt v jednom intervale a v je variancia počtu hodnôt v intervaloch. Priebeh algoritmu je potom už len taký, že generuje postupne rôzne w , kým nenájde také s najmenšou cenou $C_n(w)$.

3.3.2 Úloha histogramu vo verifikácii

Ako sme už spomenuli, primárna úloha histogramu je detailná vizualizácia distribúcie dát. Vzhľadom k vysokej úrovni detailu, nie je jednoduché porovnávanie mnoho histogramov súčasne, tak ako to bolo u krabicového diagramu. Z tohto dôvodu sa zvyčajne v praxi porovnáva navzájom len niekoľko rôznych histogramov. Vo verifikácii histogramy slúžia na porovnanie distribúcie predpovedí a pozorovaní. Porovnávanie stĺpcov histogramov sa



Obr. 3.4: Porovnanie rôznych dĺžok intervalov. Obrázok je upravený z pôvodného článku [SS07]

robí buď v dvoch rôznych diagramoch umiestnených vedľa seba alebo v jednom diagrame, kde sa stĺpce patriace k sebe umiestnia pri seba a rozdielne zafarbia, aby ich bolo možné ľahšie rozlísiť.

3.4 Čiarový diagram

Čiarový diagram je najjednoduchší spôsob ako vizualizovať jednorozmerné dátá. Využíva sa najmä na vizualizáciu spojitej premennej, keďže čiara narozenie od bodov podporuje vizuálny dojem spojitosti. Pôvod čiarového diagramu sa datuje do prapočiatkov vizualizácie informácií. Jeho autorom je zakladateľ grafických metód v štatistikе William Playfair, ktorý objavil 4 typy diagramov medzi, ktorými bol aj čiarový diagram v roku 1786 [Fri09].

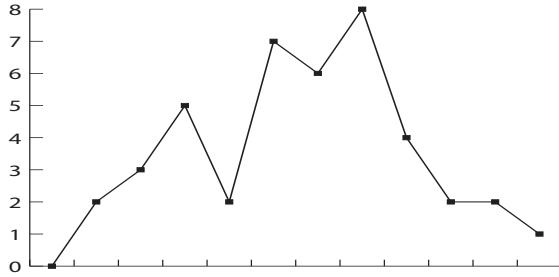
3.4.1 Konštrukcia čiarového diagramu

Pri konštrukcii čiarového diagramu sa hodnoty z množiny zobrazia v karteziánskej sústave ako body a jednotlivé dvojice susedných bodov sa následne pospájajú čiarou. Na obrázku 3.5 môžeme vidieť príklad čiarového grafu vytvoreného týmto postupom.

Čiarový graf poskytuje rôznorodé úpravy, ktoré pridávajú ďalšiu informáciu do grafu alebo zlepšujú niektoré vizuálne vlastnosti. Napríklad je možné upravovať šírku, tvar alebo farbu čiary, pridávať ďalšie čiary do grafu, zvýrazniť jednotlivé body pre dané hodnoty, použiť iný typ interpolácie medzi bodmi (zvyčajne sa používa lineárna interpolácia) a podobne.

3.4.2 Úloha čiarového diagramu vo verifikácii

Vo verifikácii je účelom čiarového diagramu vizualizácia časových radov. Na x -ovej osi zvykne byť časový interval s konkrétnymi dátumami predpovedí alebo hodiny predpovede. Rozdiel medzi nimi je ten, že konkrétny dátum je v histórii iba raz, zatiaľ čo model predpovedá pravidelne, takže predpovedné hodiny môžu byť rovnaké pre rôzne dátumy. Na y -ovej osi teda bývajú buď jednotlivé hodnoty chýb alebo pre viacero hodnôt vypočítané štatistiky ako boli spomenuté v časti 1.4.



Obr. 3.5: Príklad čiarového grafu vygenerovaného v programe Adobe Illustrator

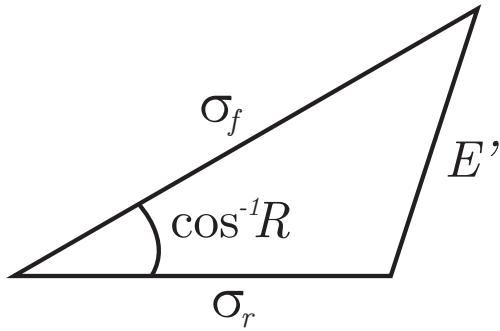
3.5 Taylorov diagram

Taylorov diagram bol špeciálne navrhnutý pre verifikáciu modelov počasia, ktorého autorom je Karl. E. Taylor [Tay01]. Podľa slov autora úlohou diagramu je štatistické zhrnutie, ako dobre si dátové vzory zodpovedajú v zmysle troch metrík: koeficient korelácie, stredná kvadratická chyba a smerodajná odchýlka. Použitie taylorovho diagramu nie je tak časté ako pri predchádzajúcich vizualizačných technikách, možno aj z dôvodu horšej čitateľnosti grafu a taktiež nemá takú veľkú tradíciu v štatistikke, ako predošlé techniky. Podobnú snahu zobraziť tri štatistiky súčasne v jednom grafe mal aj Boer a Lambert vo svojom BLT diagrame [BL01]. Tento typ grafu je veľmi podobný taylorovmu grafu, a keďže jeho použitie je ešte zriedkavejšie, tak sa mu v tejto práci nebudeme venovať.

3.5.1 Konštrukcia taylorovho diagramu

Taylorov graf porovnáva jednu alebo viacero dátových množín (f) s jednou referenčnou množinou r . Označme si hodnoty v porovnávanej množine f_n a s nimi spárované referenčné hodnoty r_n , ktoré sú definované na N diskrétnych bodoch (napríklad v čase alebo priestore). Ako sme už spomenuli, taylorov graf vizualizuje 3 štatistiky súčasne, na čo využíva ich vzájomný geometrický vzťah. Týmito štatistikami sú: koeficient korelácie R , centrovana stredná kvadratická chyba E' a smerodajné odchýlky σ_f a σ_r pre f, r . Vzorce na výpočet sú nasledovné:

$$R = \frac{\frac{1}{N} \sum_{n=1}^N (f_n - \bar{f})(r_n - \bar{r})}{\sigma_f \sigma_r}$$



Obr. 3.6: Geometrický vzťah pre popisné štatistiky $R, E', \sigma_r, \sigma_f$

, kde \bar{f} a \bar{r} sú priemerné hodnoty daných množín.

$$E' = \sqrt{\frac{1}{N} \sum_{n=1}^N [(f_n - \bar{f}) - (r_n - \bar{r})]^2}$$

Pre pripomenutie chceme upozorniť, že nejde o strednú kvadratickú chybu, ktorá bola spomenutá v časti 1.4, ale o *centrovanú* strednú kvadratickú chybu, ktorá je relatívna k stredu, teda priemeru skúmaných množín.

Rovnako pre σ_f aj σ_r je výpočet nasledovný:

$$\sigma_x = \sqrt{\frac{1}{N} \sum_{n=1}^N [(x_n - \bar{x})]^2}$$

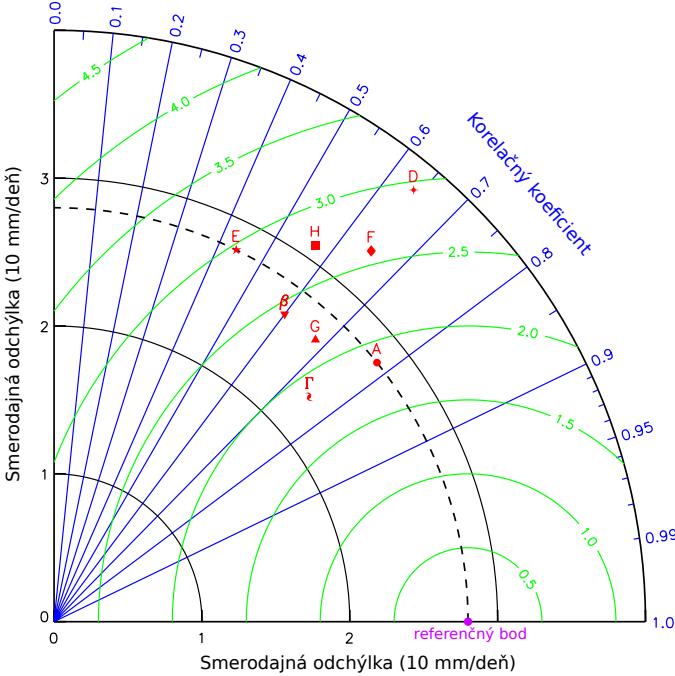
Klúčovým bodom v konštrukcii diagramu je, že s využitím týchto troch respektíve štyroch štatistik môžme skonštruovať nasledovný vzťah:

$$E'^2 = \sigma_f^2 + \sigma_r^2 - 2\sigma_f\sigma_r R$$

, ktorý sa nápadne podobá na kosínusovú vetu:

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

, kde a, b a c sú dĺžky strán trojuholníka a uhol α je protiľahlý uhol pre stranu a . Geometrický vzťah pre R, E', σ_f a σ_r je zobrazený na obrázku 3.6. Ako vidíme na obrázku 3.7,



Obr. 3.7: Taylorov diagram [Tay01]

jedna dátová množina sa pomocou týchto troch spomenutých hodnôt a vyššie uvedeného geometrického vzťahu zobrazí na grafe s polárnymi súradnicami. Azimutálna pozícia bodu nám hovorí o korelačnom koeficiente medzi množinami, preto sa vždy referenčný bod nachádza v najnižšej úrovni grafu, kde je korelácia rovná jednej. Vzdialenosť od počiatočného bodu 0 nám udáva smerodajnú odchýlku σ a vzdialenosť od referenčného bodu udáva centrovanú RMSE E' . Pre jednoduchšie odčítavanie hodnôt sa pridávajú aj označené radiálne čiary jednak od bodu 0, ale aj od referenčného bodu, ktoré sa však v praxi zvyknú vyniechať.

Popísaným spôsobom vytvoríme základný taylorov diagram, ktorý však ponúka niekoľko modifikácií [Tay05], z ktorých niektoré používané spomenieme:

- Diagram môže byť rozšírený o ďalší kvadrant vľavo, na zobrazenie zápornej kolerácie
- Štatistiky porovnávaných hodnôt môžu byť normalizované pomocou štatistik referenčných, čím dosiahneme, že na jednom diagrame možno zobraziť rôzne jednotky veličín.
- Izočiary sa vyniechávajú, aby bolo možné lepšie vidieť zobrazené body.
- Pri porovnávaní výsledkov z viacerých verzií predpovedných modelov sa zvykne na kresliť šípka medzi týmito bodmi, aby bolo lepšie vidieť vzťah medzi nimi.

- Miera E' môže byť nahradená inou. Niektoré príklady mier možno nájsť v Taylorovom článku [Tay01]
- Do grafu je možno pridať ďalšiu informáciu modifikovaním vizuálnych vlastností bodu podobne ako v bodovom diagrame. Jedným z príkladov je zobrazenie percentuálnej priemernej chyby pomocou rôznych symbolov.

3.5.2 Úloha taylorovho diagramu vo verifikácii

Tento typ diagramu bol priamo navrhnutý pre účely verifikácie predpovedných modelov počasia. Hlavným účelom diagramu je porovnávanie výkonu viacerých modelov alebo tiež jedného modelu s rôznymi nastaveniami parametrov.

Taylorov diagram skrýva dátu pod komplikovaný matematický model, ktorý nám poskytuje 3 vyššie popísané charakteristiky dát na vizualizáciu. Ak sa k tomu pridáva fakt, že diagram umožňuje iba jednu referenčnú dátovú množinu, tak použitie tohto diagramu je značne obmedzené. Využitím spomínaných troch charakteristík taktiež výrazne strácame pohľad na detail dát, ale na druhej strane nám to umožňuje jednoduché porovnávanie pomerne veľkých a komplikovaných množín na malom priestore.

Ďalšou slabinou je, že charakteristika E' sa počíta ako centrovane RMSE, ktoré je relativne vzhľadom na stred teda priemernú hodnotu dátovej množiny. Tento fakt implikuje to, že centrovane RMSE nemá jednu dôležitú vlasnosť, ktorú naopak klasické RMSE má. Touto vlastnosťou je, že čím je RMSE bližšie k nule, tým podobnejšie sú aj dátové množiny. Keďže štatistická charakteristika E' nemá túto vlastnosť, tak nám euklidovská vzdialenosť skúmaného bodu od referenčného nedáva dobrú informáciu o podobnosti dvoch množín a môže pôsobiť dezinformačne.

Kapitola 4

Návrh vizualizácie

Wolfgang Aigner vo svojej knihe *Visualization of Time-Oriented Data* [AMST11] tvrdí, že riešenie problému vizualizácie vyžaduje zodpovedanie troch otázok:

- Čo je vizualizované? - Čas a dátá
- Prečo je to vizualizované? - Požiadavky používateľov
- Ako je to vizualizované? - Vizuálna reprezentácia

Táto kapitola by mala odpovedať na všetky tieto tri položené otázky.

4.1 Charakteristika dát

V prvom rade nám do verifikácie vstupujú surové dátá, ktorými sú pozorovania a predpoveďe. Pri porovnávaní týchto dvoch dátových množín sme sa rozhodli, že ich zredukujeme na jednu množinu reprezentujúcu objektívne porovnanie predpovedí a pozorovaní so zachovaním nie všetkej, ale iba dôležitej informácie. Touto dátovou množinou je teda množina chýb predpovedí, ktorú vytvoríme tak, ako je opísané v podsekcii 1.2.3 a v sekcií 1.4.

Výsledkom tohto procesu je tabuľka chýb, kde je pre každý dátum predpovede n hodín predpovedaných hodnôt, v našom prípade 48, keďže išlo o dvojdňové predpovede. Tieto dátá v systéme ďalej spracúvame. V prvom rade ich rozdelíme na časové podintervaly podľa konfigurácie - defaultne na mesiace. Následne sa pre každý podinterval a každú hodinu predpovede vypočítajú štatistiky opísané v sekcií 1.4. Takýmto predspracovaním

sme získali dva druhy dát v podobnom ale trocha rozdielnom formáte. Ako sme spomenuli, hodnoty chýb sú v tabuľke pre každý dátum predpovede, hodnoty štatistik sú však pre časový interval predpovedí, nie len jeden konkrétny dátum.

V terminológii vizualizácie informácií hovoríme o numerických, ordinálnych a spojítých dátach, no v prvom rade ide o časovo orientované dátu. Pri tomto druhu dát je rozhodujúcou zložkou čas, ktorý opíšeme pomocou stručných charakteristík na základe už spomenutej knihy *Visualization of Time-Oriented Data* [AMST11].

Pri čase skúmame tieto charakteristiky¹:

- *Škála* (ordinálna/**diskrétna**/spojitá): Aj z pohľadu predpovedí aj z pohľadu predpovedaných hodín ide o diskrétnu škálu, keďže predpovede sú v diskrétnych časových bodoch, medzi ktorými neuvažujeme nijaké ďalšie body.
- *Rámec* (**bodový/intervalový**): Spomenuli sme, že hodnoty štatistik sú pre časový interval, preto je časový rámec intervalový. Časový rámec pre zvyšné dátu je však bodový, teda bezrozmerný okamih v čase.
- *Usporiadanie* (**lineárne/cyklické**): Dáta nás môžu navádzat k tomu, aby sme vnímali čas aj ako cyklický, keďže máme neustále opakujúce sa elementy, teda naše dvojdňové predpovede. Časové usporiadanie je však lineárne, keďže čas predpovede je úplne iný ako čas predpovedných hodín.
- *Nahliadanie* (**usporiadané/vetvené/mnoho perspektív**): Časové udalosti sa nijako nevetvia, ani nemáme dva pohľady na čas, preto ide o jednoducho usporiadany čas.
- *Zrnitosť* (**žiadna/single/multi**): Čas pre nás je rozdelený na časové jednotky, ktoré však ani nespájame, ani nerozdeľujeme na menšie. Preto hovoríme o granularite, teda zrnitosti, ale iba o single a nie multi.
- *Časové primitívi* (**okamih/interval/obdobie**): Napriek tomu, že rámec je bodový aj intervalový, jedinou časovou primitívou je okamih. Dôvodom je to, že pre každý

¹V závierke za charakteristikou sa nachádzajú možnosti delenia času na základe tejto charakteristiky. Hrubým písmom je zvýraznená vlastnosť pre naše dátu.

interval v tabuľke máme vždy iba jedinú hodnotu a nechápeme ho ako *interval* (v zmysle časovej primitívi), ale ako *okamih*.

- **Determinovanosť (determinovaný/nedeterminovaný):** Naše dáta neposkytujú žiadnu časovú nejednoznačnosť, preto sa zjavne jedná o determinovaný čas.

Autor v knihe ďalej spomína opisné charakteristiky pre časovo orientované dátá, ktoré sú tieto štyri: *škála, referenčný rámec, druh dát, počet premenných*. V stručnosti to zhrnieme tak, že naše dáta sú kvantitatívneho charakteru, abstraktné, určujúce stavy a nie udalosti s mnohými premennými. Toto uvádzame z dôvodu, že všetky tieto charakteristiky priamo vplývajú na návrh vizualizácie.

4.2 Špecifikácia požiadaviek na vizualizáciu

V tejto sekcií sa snažíme stručne zhrnúť hlavné požiadavky na vizualizáciu vyššie spomenutých dát. V prvom rade, by sme chceli zachovať základné požiadavky kladené na dobrý návrh vizualizácie vo všeobecnosti a neskôr skonkretizovať naše požiadavky pomocou *užívateľských úloh*.

4.2.1 Tri základné požiadavky

Dobrý dizajn vizualizácie sa, tak ako v iných odvetviach, snaží dosiahnuť fungujúcu rovnováhu medzi troma zložkami: *utilitas, firmitas, venustas* [MP11], čo sa z latinčiny dá voľne preložiť ako *užitočnosť, robustnosť, atraktivita*. Keďže sa v našom prípade jedná o odbornú vizualizáciu, tak sa budeme hlavne sústrediť na prvé dve požiadavky *užitočnosť* a *robustnosť*.

Užitočnosť zahŕňa funkcionality, funkčnosť, použiteľnosť a podobné pojmy. Vo vizualizácii sú tieto aspekty zhrnuté dvoma mierami - *efektnosť* (presnosť a úplnosť informácie, ktorú chcel užívateľ získať) a *efektivita* (množstvo zdrojov - zvyčajne času - potrebných na získanie žiadanej informácie). Našou úlohou pri návrhu optimálnej vizualizácie je teda pokúsiť sa, aby bola v čo najväčšej miere efektná a efektívna.

Robustnosť je pojem hovoriaci o odolnosti vizualizácie voči chybným, veľkým, rôznorodým dátam. To znamená, že ako dobre a akým spôsobom sa vysporadúva s týmito problémami. Taktiež hovorí o správnosti vizualizácie, teda či berie do úvahy ľudský percepčný systém a nejakým spôsobom nedodáva klamlivú informáciu. Hovoríme o javoch, ktoré poznáme zvyčajne z optických ilúzií. Príkladom môže byť vnímanie perspektívy, kontextuálne vnímanie farieb, vnímanie hodnôt na základe plochy a nie dĺžky a iné podobné javy, na ktoré je nutné si dávať pozor.

Atraktivita alebo tiež estetika - krása daného riešenia. Tento pojem nie je limitovaný vizuálnou stránkou vizualizácie, ale taktiež nejasným aspektom, akými sú originalita, inovácia a novosť. Článok od Moera a Purchaseho [MP11] tvrdí, že tento aspekt je pri-veľmi podceňovaný, no i my sa radíme medzi tých, ktorí nekladú naň taký dôraz, ako na predošlé dva.

4.2.2 Užívateľské úlohy

Konkrétnejšie požiadavky na vizualizáciu sa najlepšie špecifikujú pomocou užívateľských úloh. V tejto časti pomocou *taxonómie úloh vizualizácie* navrhnutej súrodencami Andrienko [AA05], zhrnieme základné uvažované úlohy, ktoré budú užívatelia vykonávať. Jednotlivé úlohy sa definujú pomocou ukážkových otázok, ktoré sa budú klásiť na vizualizáciu.

Elementárne úlohy

Tieto úlohy sú mierené na individuálne dátové elementy, teda napríklad konkrétné hodnoty.

- *Vyhľadávanie*
 - Aké veľké je RMSE a MFE pre piatu hodinu predpovede?
 - Aká je distribúcia chýb pre túto hodinu?
- *Inverzné vyhľadávanie*
 - Ktorá predpovedaná hodina mala pre január najväčšiu priemernú chybu?

- Ktoré predpovede sa výrazne líšia od ostatných?
 - Ktoré dni v roku bola priemerná chyba najnižšia?
 - Ku ktorej predpovedi patrí outlier.
- *Porovnávanie*
 - Je chyba nižšia v prvých alebo neskorších hodinách?
 - Aký je rozptyl chýb na začiatku a na konci predpovede?
 - *Inverzné porovnávanie*
 - Je chyba pre daný mesiac najnižšia okolo desiatej hodiny?
 - *Hľadanie vzťahov*
 - Aký vplyv má outlier na priemernú chybu predpovede?

Synoptické úlohy

Narozdiel od elementárnych úloh, synoptické alebo skôr prehľadové úlohy zahŕňajú celkový pohľad na dátu, teda nie na konkrétné hodnoty, ale na množiny hodnôt.

- *Vyhľadávanie (Definovanie vzorov)*
 - Existujú nejaké výkyvy v chybách z ktorých sa počítali štatistiky?
 - Aké sú trendy v magnitúde chýb počas roka?
 - Nadhodnocuje, či podhodnocuje model predpovede?
- *Inverzné vyhľadávanie (Vyhľadávanie vzorov)*
 - Ktorú predpovednú hodinu sú predpovede najpresnejšie?
 - Ktoré ročné obdobie sú predpovede najslabšie?
- *Porovnávanie vzorov*
 - Porovnaj správanie modelu pre rôzne stanice.
 - Porovnaj správanie modelu pre rôzne mesiace.
- *Inverzné porovnávanie vzorov*

- Je správanie modelu rozdielne v prvý deň predpovede ako v druhý?
- *Hľadanie vzťahov*
 - Aký vplyv majú horúcejšie dni na úspech predpovede?
 - Aký vplyv má rozptyl chýb na úspešnosť predpovede?

4.3 Návrh vizualizácie štatistik verifikácie

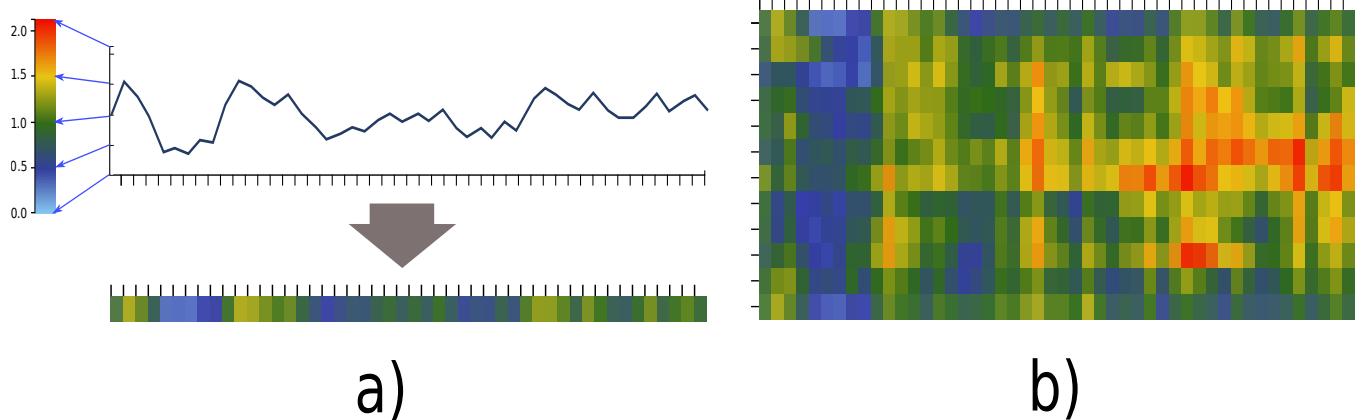
Štatistiky verifikácie sú dátia v časových radoch. V sekcii 3.4 sme uviedli, že bežne používaným a veľmi účinným spôsobom vizualizácie časových radoch je čiarový diagram. V našej práci sme sa odrazili od tohto faktu a navrhli sme dva pohľady na dátia. Jednako je to prehľad jedného druhu štatistik pre danú premennú počas skúmaného intervalu a druhým pohľadom je detail jedného časového podintervalu s rôznymi štatistikami alebo premennými.

4.3.1 Prehľad štatistik (*Farebná mapa*)

Ako sme spomenuli vyššie, na vizualizáciu časových radoch sa bežne používa čiarový diagram. Z našej skúsenosti vieme, že táto technika nie je vhodná pre porovnanie mnohých chronologicky usporiadaných radoch súčasne. Problémom je, že ak umiestnime viacero radoch do jedného grafu, tak strácame informáciu o chronologickom poradí. Ak však umiestnime rady do jednotlivých grafov, tak narážame na problémy s nedostatočným priestorom alebo s čitateľnosťou grafu.

Medzi techniky na vizualizáciu viacerých časových radoch súčasne patria *prepletený graf* (*braided graph*), *malé násobky* (*small multiples*) alebo *horizon graf* [JME10]. My sme vzhľadom na vyššie spomenuté problémy zvolili pomerne jednoduché a účinné riešenie, zobraziť štatistiky na iný vizuálny parameter, konkrétnie na farbu. Jednotlivé hodnoty sa teda v grafe zobrazia ako tesne za sebou umiestnené farebné obdĺžniky², ktoré vytvoria jeden celistvý pás (pozri porovnanie s čiarovým diagramom na obrázku 4.1a. Umiestnením

²Experimentovali sme taktiež s inými tvarmi (kruhy a obdĺžniky so zaoblenými hranami), ale aj s rôznymi veľkosťami medzier medzi tvarmi. Použité nastavenie sa však ukázalo ako najvhodnejšie pri porovnávaní hodnôt vo vertikálnom aj horizontálnom smere.



Obr. 4.1: a) Konštrukcia jedného pásu zobrazením hodnôt grafu na farebnú škálu b) Výsledná vizualizácia, ako farebná mapa

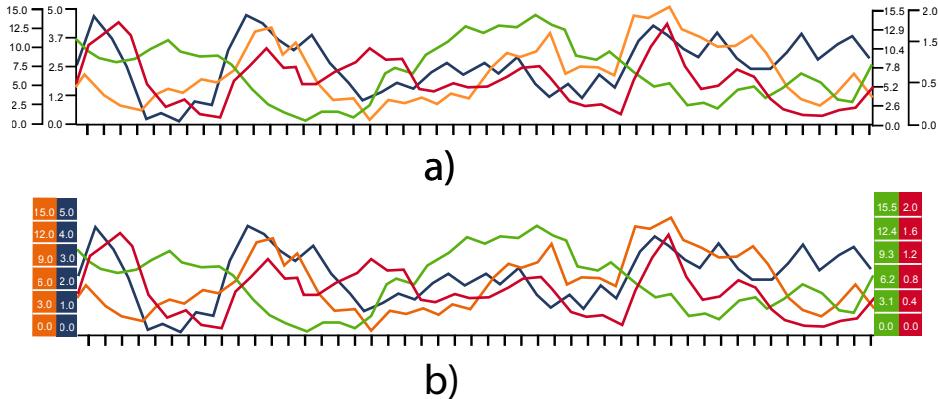
takýchto pásov pod seba vytvoríme farebnú mapu štatistik. Výslednú vizualizáciu možno vidieť na obrázku 4.1b.

V našej aplikácii sa tento spôsob vizualizácie použil v dvoch prípadoch. Prvým prípadom bolo zobrazenie mesačných štatistik, pre 48 hodinové predpovede. Teda sme mali 12 mesiacov v roku a pre každý mesiac 48 hodnôt, čo sa nám zobrazilo ako farebná mapa 12×48 (pozri obrázok 4.1b). Druhým prípadom bolo zobrazenie chýb pre konkrétnu predpovede, ktorých mohlo byť variabilný počet podľa nastavení systému. Zvyčajne však išlo približne o 30 predpovedí pre každý mesiac.

4.3.2 Detail štatistik (*Mnoho-čiarový diagram*)

Napriek tomu, že zobrazením hodnôt na farbu nestrácamo žiadnu informáciu, je dokázané, že človek nedokáže tak dobre porovnávať farby, ako napríklad vzdialenosť. Preto predošle spomenutá technika je súčasne vhodná na preskúmanie celého intervalu ako celku, avšak nie na detailné porovnávanie hodnôt v čase. Z tohto dôvodu sme sa rozhodli použiť taktiež spomínaný čiarový diagram.

Tento typ diagramu poskytuje dostatočne veľký vizuálny priestor na to, aby sme doň mohli pridať ďalšie dátu. Rozhodli sme sa preto umožniť užívateľovi vizualizovať bud' rôzne druhy štatistik súčasne pre danú predpoved' a veličinu (napr RMSE, MAE, MFE), alebo tiež rôzne veličiny (napr. tlak, teplotu, zrážky) pre túto predpoved'. V prvom prípade nám postačí jednotná škála, keďže sa hodnoty zväčša nachádzajú v podobnom rozmedzí.



Obr. 4.2: a) Bežné zobrazenie viacerých škál b) Nami navrhnuté zobrazenie viacerých škál

V druhom prípade je potrebných viaceré škály, keďže napríklad chyby teploty sa pohybujú v rozmedzí 5 stupňov, zatiaľ čo chyby tlaku sa pohybujú v rozmedzí stoviek. Ak by sme tieto dve veličiny vizualizovali spolu, tak krivka teploty by bola sploštená natol'ko, že by nebolo možné z nej odčítať hodnoty.

Vytvorenie mnoho-čiarového diagramu s jednotnou škálou je jednoduché, keďže sa na dizajne grafu nič nemení, len sa pridá ďalšia krivka. Pri viacerých rôznych škálach sa musia zobraziť aj tieto škály. Zvyčajne sa umiestňujú striedavo vľavo a vpravo, postupne za sebou smerom od stredu grafu. Problém však spočíva v tom, že pri väčšom počte nemôžme určiť, ktorá škála patrí ku ktorej krivke a s postupujúcou vzdialenosťou škály od grafu sa znižuje schopnosť užívateľa odčítavať hodnoty zo škály. V našej práci sme navrhli jednoduché spôsob, akým sme sa pokúsili vyriešiť oba problémy súčasne.

Naše riešenie spočíva v tom, že každá škála sa zobrazí ako obdĺžnik vyplnený farbou krivky, ktorá prislúcha k danej škále. Taktiež čiaročky indikujúce hodnoty v danom bode sa zobrazia pravidelne a v rovnakej vzdialosti pre každú škálu.

Na obrázku 4.2 môžme vidieť porovnanie bežne používaného spôsobu zobrazenia škál a nášho spôsobu. Náš spôsob jednoznačne priraduje, pomocou farby, krivky ku škálam a jednotné, pravidelné rozmiestnenie indikátorov hodnôt nám rieši problém pri odčítavaní hodnôt, keďže nie je potrebné sa dívať na vzdialenosť škály.

4.4 Návrh vizualizácie distribúcie chýb

Pri verifikácii predpovede spojitej premennej sme použili štatistické metódy spomenuté v sekciu 1.4, ktorých výsledok sme následne vizualizovali. Pôvodné dáta však zostali skryté za použitým matematickým modelom, a tak sme stratili informáciu o distribúcii chyby. Pri verifikácii sa štandardne používajú tri metódy na priamu, či nepriamu vizualizáciu a analýzu distribúcie, ktoré sme opísali v kapitole 3. Týmito metódami sú bodový graf (pozri sekciu 3.1), krabiový diagram (pozri sekciu 3.2) a histogram (pozri sekciu 3.3).

My sme sa rozhodli napasovať vizualizáciu priamo na dátu verifikácie a pri jej návrhu sme vyskúšali niekoľko vizualizačných techník a zvážili ich silné a slabé stránky.

4.4.1 Graf hustoty

Jedným z viacerých spôsobov, ako pomerne presne určiť distribúciu chýb je pomocou *grafu hustoty*. Ten sa skonštruuje jednoducho z *funkcie hustoty*, ktorú získame *odhadom hustoty* z dát.

Prvý pohľad na dátu by nám vravel, že ide o dvojrozmerné dátu a teda je potrebné použiť odhad hustoty dvoch premenných. Takýto postup by samozrejme bol možný, ale dovedol by nás k chybnej vizualizácii a tak aj k mylnej predstave o dátach. Dôvodom je to, že máme záujem o analýzu distribúcie chýb pre každú hodinu predpovede zvlášť, čo znamená, že chceme zistiť distribúciu iba v jednom smere.

Pre vytvorenie grafu hustoty je v prvom rade potrebné vybrať správny spôsob odhadu hustoty. Jedným z bežne používaným štandardným spôsobom je odhad hustoty pomocou jadra, po anglicky známy ako *kernel density estimation* (KDE) [Ros56, Par62]

Nech máme n hodnôt $x_i, 1 \leq i \leq n$, z ktorých chceme určiť odhad hustoty, potom estimátor hustoty $\hat{f}_h(x)$, ktorý approximuje *funkciu hustoty pravdepodobnosti* (PDF) f , sa vypočíta takto:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=0}^n K\left(\frac{x - x_i}{h}\right)$$

kde h je šírka jadra a $K(x)$ je funkcia jadra (skrátene iba jadro), ktorá by mala splňať

nasledovné vlastnosti:

$$K(x) \geq 0$$

$$\int K(x)dx = 1$$

tieto vlastnosti hovoria, že $K(x)$ je na celom definičnom obore nezáporná a jej integrál je rovný 1, teda sa jedná o normalizovanú funkciu. Bolo preštudovaných mnoho jadier, ako napríklad uniformné, tri-angulárne, Epanechnikovo [Tur93], kvadratické, Gaussové, kosínusové a veľa ďalších. Najbežnejšie a zrejme aj najpraktickejšie [MS93] je Gaussovo (normálne) jadro, ktoré sme použili aj my:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Volba jadra však nemá na výsledok až taký vplyv, ako volba šírky jadra h . My sme použili výpočet šírky jadra na základe dátovej množiny, ktorý approximuje optimálnu šírku jadra [Sco92, BA97]³. Všeobecne pre d dimenzionálne dát je vzorec nasledovný:

$$h = \sigma \left(\frac{4}{(d+2)n} \right)^{\frac{1}{d+4}}$$

kde σ je smerodajná odchýlka vypočítaná z daných dát a n je veľkosť dátovej množiny. V našom prípade je $d = 1$, a tak sa nám vzorec zjednoduší na

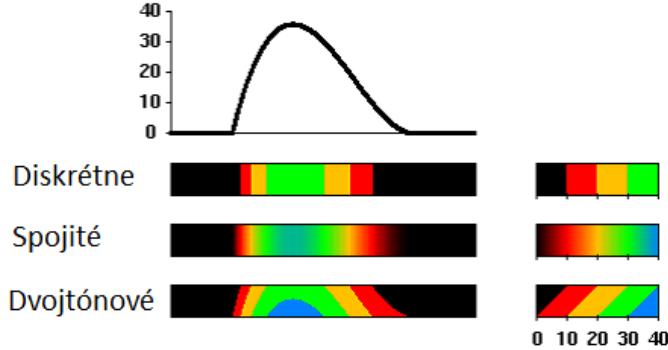
$$h = \sigma \left(\frac{4}{3n} \right)^{\frac{1}{5}}$$

Aby sme zjednodušili výpočet, tak sme si konštanty vypočítali predom a zaokrúhlili na 2 desatinné miesta, čo považujeme za dostačujúce. Výsledný vzorec, ktorý sa nakoniec objavil v aplikácii je takýto:

$$h = 1.06 \times \sigma \times n^{-\frac{1}{5}}$$

Takýmto spôsobom sme si pre každú hodinu predpovede určili samostatnú funkciu $\hat{f}_h(x)$, ktorú môžme vizualizovať. Zvyčajne sa funkcie hustoty vizualizujú ako bežné funkcie, teda pomocou čiarového diagramu, tak ako na obrázku 3.5. V našom prípade by bol

³Optimálna šírka jadra je taká, ktorá minimalizuje strednú integrovanú kvadratickú chybu (MISE).



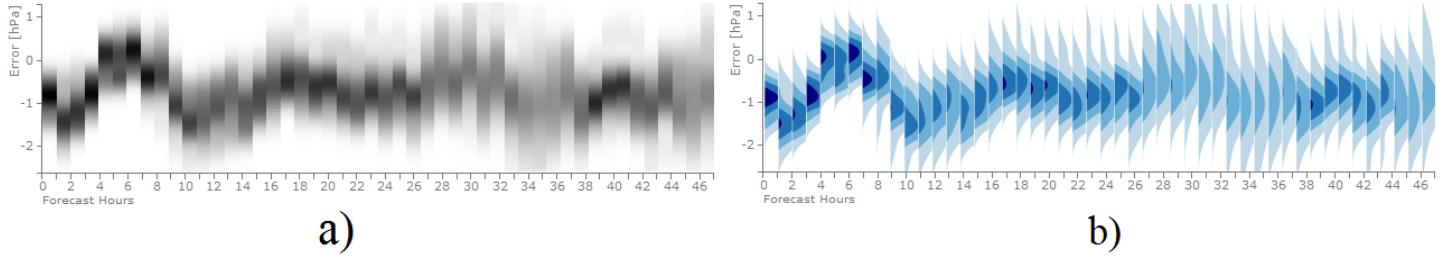
Obr. 4.3: Porovnanie dvoch konvenčných techník farbenia so Saitovým dvojtónovým farbením. Obrázok pochádza z pôvodného článku [SMY⁺05].

tento prístup nepraktický, keďže máme veľké množstvo funkcií, tak jednak by bola takáto vizualizácia nepraktická pri porovnávaní distribúcií a taktiež na to nemáme potrebný vizuálny priestor.

Opäť sme teda zvolili štandardné riešenie, ako ušetriť vzácný priestor a to tak, že hodnoty, ktoré by boli zobrazené na y -ovú os zobrazíme na zvolenú farebnú škálu. Vďaka tomu by teoreticky mohol mať graf hustoty pre jeden čas predpovede šírku 1 pixel bez straty akejkoľvek informácie.

Vieme, že pre ľudí nie je také jednoduché pozorovať malé rozdiely medzi dvoma farbami. Testovaním sme zistili, že tento fakt spôsobuje problémy aj pri našej vizualizácii, kedy sa pre pozorovateľa strácajú jemné výkyvy hodnôt. Riešením by bolo nepoužiť spojité, ale diskrétné farbenie, ktoré rozdeľuje interval hodnôt na podintervaly, ku ktorým priradí konkrétnu farbu (pozri obr. 4.3). Výhodu je, že takéto farbenie má výraznejšie farebné rozdiely. Týmto spôsobom by sme však stratili privel'a informácie a nebolo by potrebné použiť na odhad hustoty KDE, ale postačoval by aj histogram.

Riešenie tohto problému sme našli v práci od Takafumi Saita z roku 2005 [SMY⁺05]. V tomto článku autor so svojimi kolegami navrhuje nový spôsob pseudo farbenia, ktoré nazýva *two-tone pseudo coloring* alebo inak po slovensky *dvojtónové pseudo farbenie*. Navrhnutá metóda priraduje každej hodnote intervalu dve výrazne rôzne farby resp. farebné časti. Toto sa uskutočňuje podobne ako pri spojiteľnom farbení, avšak sa neinterpolujú farby, ale pomer veľkostí dvoch zafarbených častí. Na obrázku 4.3 možno vidieť porovnanie troch spomenutých techník farbenia.



Obr. 4.4: Porovnanie použitia dvoch techník farbenia. a) Spojité pseudo farbenie b) Dvojtónové pseudo farbenie

Na obrázku 4.4 vidno porovnanie vizualizácií s použitím spojitého a dvojtónového farbenia.

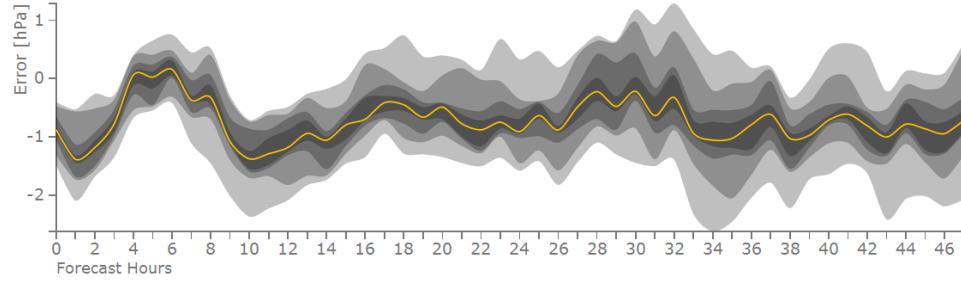
4.4.2 Pruhový kvantilový diagram

Z časti 3.1.2 sme už dobre oboznámený s pojmom kvantil. Klasický kvantilový diagram zobrazuje kvantil hodnôt pre jednu dimenziu. Ak si vezmeme naše dátá, kde pre každý čas je niekoľko chýb predpovedí, tak kvantilový diagram skonštruujeme tak, že pre každý čas vypočítame kvantil, ktorý zobrazíme ako bod alebo ako súčasť lomenej čiary v grafe. Prirodzeným rozšírením je zobrazovať nielen jeden kvantil, ale mnoho kvantilov súčasne. Zvyčajne sú to kvantily $Q_{0.25}, Q_{0.75}, Q_{0.5}$, teda kvartily z ktorých sa konštruuje krabíkový diagram, a taktiež kvantily $Q_{0.02}, Q_{0.98}$. V článku od Bade et al. [BSM04] ja navrhnutý spôsob rozšírenia krabíkového diagramu odstránením krabíc a ponechaním kvartilov a extrémov.

V našej práci sme využili toto rozšírenie na lepšie zobrazenie distribúcie a navrhli sme takzvaný *pruhový kvantilový diagram*, ktorý sa neobmedzuje iba na kvartily.

Jeden *pruh* v grafe definujeme pomocou dvojíc hodnôt v čase - spodným a jeho protilehlým kvantilom. Spodným kvantilom je kvantil Q_α a k nemu protilehlý je kvantil $Q_{1-\alpha}$, kde $0 \leq \alpha \leq \frac{1}{2}$. Vidíme teda, že pruh ohraničuje hodnoty v okolí stredu usporiadanej množiny dát. Špeciálnym prípadom pruhu je pre $\alpha = \frac{1}{2}$, vtedy spodný aj horný kvantil je $Q_{0.5}$, čo je vlastne medián.

Pri návrhu vizualizácie sme sa snažili, aby mohol mať diagram variabilný počet pruhov a taktiež, aby rozostup medzi pruhmi bol pravidelný. Pri riešení tohto problému, sme sa



Obr. 4.5: Pruhový kvantilový diagram so 4 pruhmi a extrémami. Medián je zvýraznený žltou farbou.

inšpirovali krabicovým diagramom, kde sa hodnoty delia mediánom na dve časti, ktoré sa ďalej tak tiež delia ich mediánom. Ide teda o rekurzívne delenie usporiadanej množiny na polovicu do hĺbky 2. Túto myšlienku sme rozšírili na ľubovoľnú hĺbku delenia d . Potom i -ty pruh $\mathcal{P}_d(i)$ pre hĺbku d definujeme takto:

$$p_d(i) = (Q_\alpha, Q_{1-\alpha}), \alpha = i \times (0.5)^d$$

$$\mathcal{P}_d(i) = \{(t, p_d(i)) : t \in I\}$$

a množina všetkých pruhov grafu pre hĺbku delenia d je definovaná takto:

$$\{\mathcal{P}_d(i) : 0 \leq i \leq 2^{d-1}, i \in \mathbb{N}\}$$

Aby sme sa vyhli rekruzii, vypočítali sme si krok medzi susednými kvantilmami pri hĺbke d , ktorý je $(0.5)^d$, a jednotlivé pruhy sme generovali s týmto krokom. Pri rekruzívnom delení sa vygeneruje 2^d kvantilov a z nich je možné vyrobiť 2^{d-1} pruhov, preto sme index i obmedzili na $i \leq 2^{d-1}$. Z tohto vidíme, že počet pruhov grafe s rastúcou hĺbkou rastie exponenciálne, preto odporúčame, aby d bolo maximálne 4, kedy sa nám množina rozdelí na 16 častí 15 hexadecimí a tak vznikne 8 pruhov.

Na obrázku 4.5 vidíme, že takto definovaný pruh sa potom vizualizuje, ako plocha medzi krivkami, ktoré tvoria dvojice hodnôt patriace danému pruhu, s výnimkou špeciálneho prípadu $Q_{0.5}$, ktorý vizualizujeme ako krivku.

4.4.3 Funkčný krabicový diagram

Pre pochopenie dát je dôležité, aby sme sa vedeli pozrieť na hodnoty v ich kontexte. Všetky predošlé techniky uvažovali o chybe ako o samostatnej hodnote pre určitý predpovedný čas, avšak chyby sa nenachádzajú len v kontexte predpovedného času, ale aj v kontexte konkrétnej predpovede. Preto môžme uvažovať o predpovediach ako o funkciách $x_i(t)$, kde $i \in \{1..n\}$ je poradie predpovede a $t \in I$ je čas predpovede, kde I je časový interval predpovede z \mathbb{R} (v našom prípade sa jednalo o dvojdňovú, teda 48 hodinovú predpoved').

Takýmto spôsobom sme sa dostali do novej situácie, kedy nechceme vizualizovať distribúciu jednotlivých chýb, ale celých predpovedí, ktoré chápeme ako funkcie. Na riešenie tohto problému existuje niekoľko spôsobov, z ktorých sme si zvolili *funkčný krabicový diagram* [SG11], keďže myšlienkovo vychádza z klasického krabicového diagramu, ktorý je jednak na túto situáciu vhodný ale je tiež medzi užívateľmi dobre známy a zaužívaný.

Ako sme spomenuli v časti 3.2, klasický krabicový diagram potrebuje na svoju konštrukciu 5 hodnôt: 3 kvartily a 2 extrémy. Aby sme tieto hodnoty našli pre funkcie, musíme ich vedieť porovnať a povedať, ktorá je "väčšia" alebo "menšia". Autori funkčného krabicového diagramu riešia problém s využitím takzvanej pásmovej hĺbky (*band depth*) [LPR09]. Grafom G funkcie x je množina bodov $G = \{(t, x(t)) : t \in I\}$. Pásma \mathcal{B} (*band*) v \mathbb{R}^2 ohraničené krivkami $x_{i_1}, x_{i_2}, \dots, x_{i_k}$, kde $k \geq 2$ je definované takto:

$$\mathcal{B}(x_{i_1}, x_{i_2}, \dots, x_{i_k}) = \{(t, y) : t \in I, \min_{r=1..k} x_{i_r}(t) \leq y \leq \min_{r=1..k} x_{i_r}(t)\}$$

Pásma \mathcal{B} (pozri obrázok 4.6) je teda množina všetkých bodov existujúcich medzi extrémami všetkých kriviek, ktoré doň vstupujú ako parameter. Pomocou týchto dvoch funkcií môžme definovať pomocnú funkciu $BD_n^{(j)}(x)$ pre krivku x , ktorá vyzerá takto:

$$BD_n^{(j)}(x) = \binom{n}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \dots < i_j \leq n} \mathcal{I}\{G(x) \subset \mathcal{B}(x_{i_1}, \dots, x_{i_j})\}, j \geq 2$$

kde j je počet kriviek definujúcich pásmo \mathcal{B} , n je celkový počet kriviek a \mathcal{I} je takáto

funkcia:

$$\mathcal{I}(x) = \begin{cases} 1 & \text{ak platí } x \\ 0 & \text{ak neplatí } x \end{cases}$$

Pomocná funkcia BD pre krivku x definuje pomer všetkých pásem zložených z j kriviek, v ktorých sa graf $G(x)$ nachádza, ku všetkým možným j -ticiam kriviek vybraným z n . Samotná funkcia pásmovej hĺbky \mathcal{BD} pre krivku x je definovaná takto:

$$\mathcal{BD}_{n,J}(x) = \sum_{j=2}^J BD_n^{(j)}(x), J \geq 2$$

Hĺbka pásma \mathcal{BD} je teda suma všetkých BD pre počet kriviek 2 až J .

Autor článku definujúci pojem pásmová hĺbka navrhol taktiež flexibilnejšiu verziu s použitím pomocnej funkcie MBD (*modified band depth*) [LPR09]. V pravom rade je potrebné zadefinovať si funkciu A , ktorá určí všetky časové body, kedy sa krivka x nachádza v pásme B .

$$A(x, B) = \{t \in I : (t, x(t)) \in G(x) \wedge (t, x(t)) \in B\}$$

V spomínanom článku autori využívajú alternatívnu definíciu funkcie A , do ktorej vstupuje $j + 1$ kriviek. Jej význam zostáva rovnaký ako pri našej definícii, avšak náš prístup považujeme za jednoduchší a zrozumiteľnejší. S využitím *Lebesgueovej miery* λ autori ďalej definujú funkciu λ_r , ktorá nám dáva "pomer času, ktorý krivka strávi v pásme":

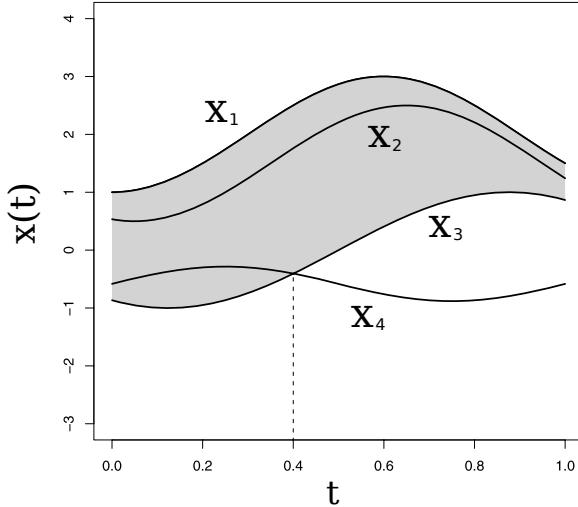
$$\lambda_r(A) = \frac{\lambda(A)}{\lambda(I)}$$

Nová pomocná funkcia MBD je definovaná nasledovne:

$$MBD_n^{(j)}(x) = \binom{n}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \dots < i_j \leq n} \lambda_r\{A(x, \mathcal{B}(x_{i_1}, \dots, x_{i_j}))\}, j \geq 2$$

Ak platí, že $G(x) \subset \mathcal{B}(x_{i_1}, \dots, x_{i_j})$, tak funkcia MBD sa degeneruje na BD [SG11]. Na obrázku 4.6 môžeme vidieť krivku x_4 , ktorá by bola v prípade BD započítaná s hodnotou 0, zatiaľ čo flexibilnejšia metóda MBD , by ju započítala ako 0.4.

V našej aplikácii sme sa rozhodli, že budeme pásmo definovať pomocou iba dvoch



Obr. 4.6: Na obrázku je znázornený príklad pásma $\mathcal{B}(x_1, x_3)$ zloženého z 2 funkcií x_1, x_3 . Taktiež môžeme vidieť funkciu x_2 , ktorej graf leží v pásme, zatiaľ čo x_4 podľa BD neleží vôbec a podľa MBD len čiastočne.

kriviek, čo nám vzorec výrazne zjednodušilo. Taktiež to implikovalo fakt, že pri výpočte MBD nie je potrebné $\binom{n}{j}^{-1}$, keďže berieme pásma zložené vždy z rovnakého počtu kriviek. Pre naše účely sme si taktiež zjednodušili funkciu λ_r na λ , keďže nepotrebuje vlastnosť tejto funkcie, ktorá dosahovala to, že MBD pre špeciálny prípad degeneruje na BD . Po týchto úpravách výsledný vzorec pre naše \mathcal{BD}' vyzerá nasledovne:

$$\mathcal{BD}'_n(x) = \sum_{1 \leq i_1 < i_2 \leq n} \lambda \{ A(x, \mathcal{B}(x_{i_1}, x_{i_2})) \}$$

Teraz, keď sme úspešne definovali mieru, podľa ktorej môžeme usporiadať funkcie resp. ich krivky, je veľmi ľahké skonštruovať funkčný krabicový diagram.

Nech sú naše funkcie predpovedí x_1, \dots, x_n usporiadane zostupne podľa \mathcal{BD}' . Potom krivka pre funkciu x_1 má najvyššiu pásmovú hĺbku a predstavuje strednú hodnotu pre množinu funkcií, teda niečo podobné ako medián hodnôt pri krabicovom diagrame. Pri konštrukcii funkčného krabicového diagramu nám taktiež pomôže koncept centrálneho regiónu [LPS99]. Centrálny regón C pre 50% kriviek je pásmo vytvorené z 50% najhlbších kriviek, teda:

$$C_{0.5} = \mathcal{B}(x_1, x_2, \dots, x_{\lceil n/2 \rceil})$$

Vidíme, že Centrálny región $C_{0.5}$ zaobala 50% najhlbších kriviek, a teda sa jedná o akúsi analógiu pre medzikartilový rozsah (IQR) v klasickom krabikovom diagrame (pozri sekciu 3.2), ktorý ohraničoval 50% centrálnych dát. Zobrazením hraničných bodov tohto regiónu získame obálku myšlienkovo totožného boxu v krabikovom diagrame (pozri obrázok 4.7b). Táto myšlienka sa dá použiť ďalej a môžme, tak ako na obrázku 4.7c, zobraziť tiež 25%-ný a 75%-ný centrálny región $C_{0.25}$, $C_{0.75}$, avšak kvôli nižšej čitateľnosti grafu sme túto alternatívu nepoužili.

V prípade, že nepotrebujeme identifikovať *outlier*-ov, tak extremálne hodnoty je už veľmi jednoduché získať, pretože ich tvorí pásmo zložené zo všetkých kriviek $B(x_1, \dots, x_n)$. V opačnom prípade musíme najprv identifikovať *outlier*-ov, ktorých potom vylúčime z výpočtov. Opäť sa využíva myšlienka z klasického krabikového diagramu, kedy sa *outlier* určil pomocou hodnoty $c \times IQR$, kde c bolo zvyčajne 1.5. Hranice sa teda získajú naškálovaním centrálneho regiónu so škálovacím faktorom 1.5 a všetky krivky, ktoré sa v tomto regióne nenachádzajú, budú považované za *outlier*-ov. Test na *outlier*-a vyzerá teda takto:

$$y_i(t) = 1.5 \times x_i(t)$$

$$\text{isOutlier}(x) = [G(x) \notin B(y_1, \dots, y_{\lceil n/2 \rceil})]$$

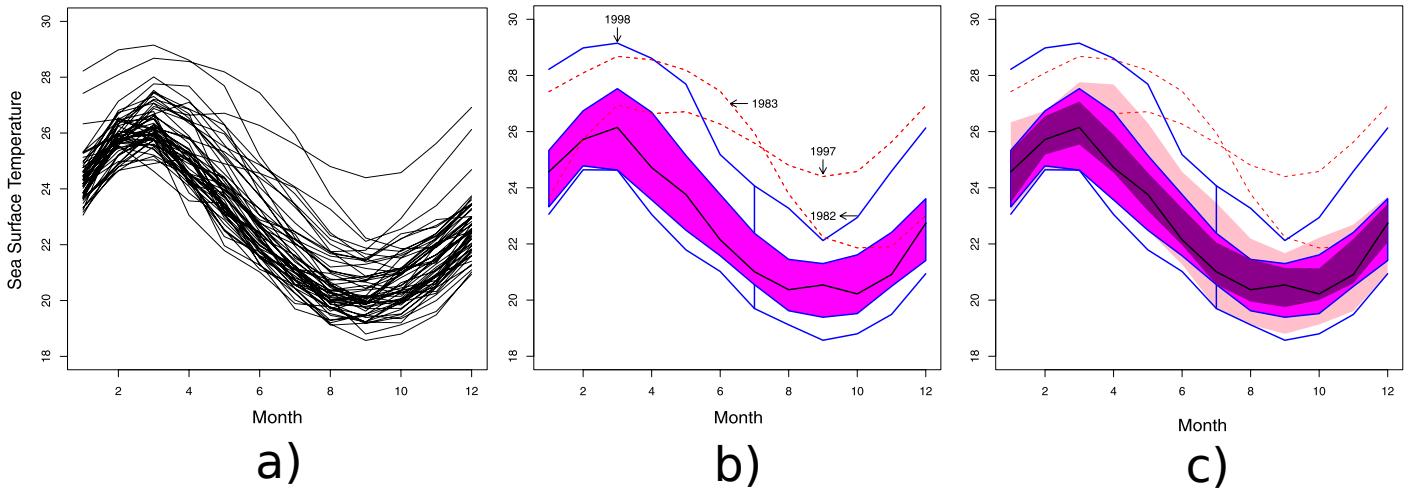
Na obrázku 4.7b môžme vidieť červené prerušované čiary, ktorými sú *outlier*-i znázornené.

V našej práci sme do diagramu pridali ešte jednu krivku pre ľubovoľnú štatistiku vypočítanú z chýb ako napríklad MFE, MAE alebo RMSE (pozri sekciu 1.4). Takto môžme spraviť porovnanie distribúcie chýb s vypočítanou štatistikou.

4.4.4 Mapa distribúcií

Návrh tejto metódy spočíva v tom, že vizualizácia prehľadu štatistik pomocou farebnej mapy ponecháva ešte dostatok nevyužitého vizuálneho priestoru. Obsah obdĺžnikov, ktoré tvoria túto mapu, sa dá zaplniť ďalším vizuálnym elementom, ktorým je možné dodať informáciu o distribúcii.

Malý priestor však neponúka príliš veľa možností na veľký detail. Rozhodli sme sa preto poskytnúť užívateľovi informáciu o distribúcii pomocou jedinej hodnoty, konkrétnie



Obr. 4.7: Obrázky z článku *Functional Boxplots* [SG11] a) Funkcie meraní teploty hladiny mora b) Funkčný krabikový diagram c) Rozšírený Funkčný krabikový diagram o centrálné regióny $C_{0.25}$ a $C_{0.75}$

štandardnej odchýlky σ .

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_i)^2}$$

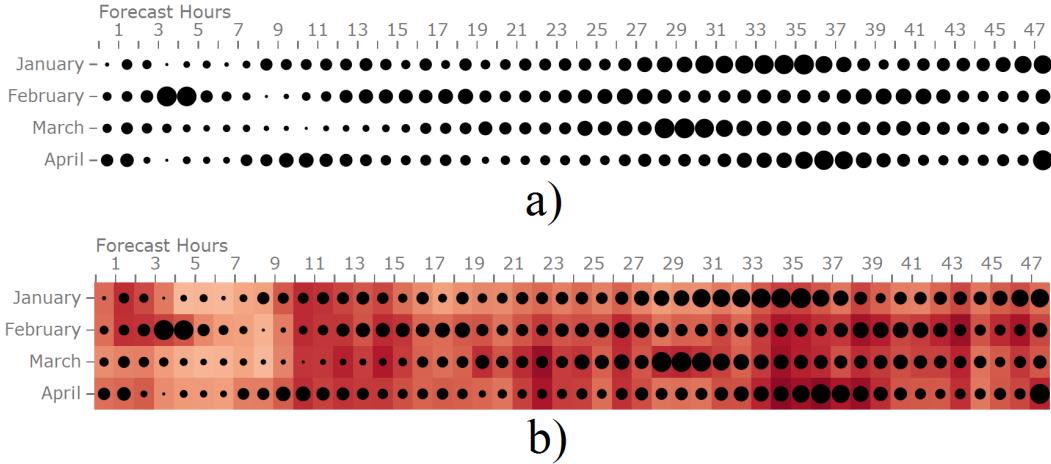
Jednotlivé hodnoty sa zobrazia vo vizualizácii ako kruhy rôznej veľkosti v závislosti od σ . Ľudský percepčný systém vníma veľkosť kruhu v zmysle plochy, preto nezobrazujeme hodnoty σ priamo na veľkosť polomeru r , ale na obsah kruhu. Vzťah, ktorým vypočítame polomer je odvodený od vzorca pre obsah kruhu a vyzera nasledovne:

$$r = \sqrt{\frac{S}{\pi}}$$

$$S = k\sigma$$

kde k je koeficient škálovania. Nemusíme sa báť záporných hodnôt pod odmocninou, keďže σ nemôže nadobúdať záporné hodnoty a $k > 0$.

Takýmto spôsobom dodáme užívateľovi rýchly prehľad o tom, ako môže štatistikám z chýb dôverovať, aké sú trendy alebo vzory v disperzii a podobne. Tento typ vizualizácie sa dá kombinovať s farebnou mapou, ale môže existovať aj samostatne podľa preferencií užívateľa. Na obrázku 4.8 je znázornený príklad výslednej vizualizácie.



Obr. 4.8: Mapa distribúcií a) Bez farebnej mapy b) Spolu s farebnou mapou

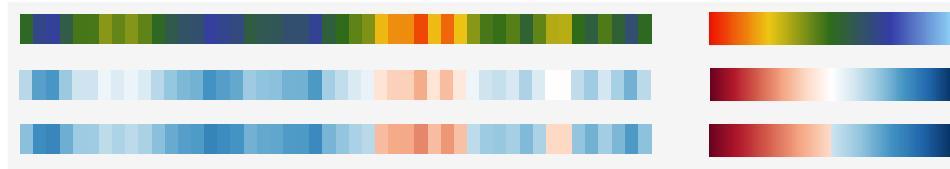
4.5 Návrh farebnej palety

Farba je pravdepodobne najviac podceňovaný a nesprávne používaný vizuálny parameter vo vizualizácii dát. V tejto časti opisujeme návrh farebnej palety pre vizuallizáciu pomocou farebnej mapy (pozri sekciu 4.3.1).

V prvom momente sme siahli po veľmi prvoplánovom riešení a použili sme takzvanú *dúhovú farebnú škálu*, ktorú možno vidieť v sekcií 4.3.1 na obrázku 4.1. Výskum ukázal a mnohí vedci sa na tom zhodujú, že tento typ farebnej palety je len zriedkakedy najlepšia voľba pri vizualizácii. Dúhové farebné palety majú niekoľko slabých stránok, ktoré zhŕňa David Borland a Russell M. Taylor II vo svojom článku [BI07].

Prvou slabinou je, že farby v dúhovej palete, narozené od monochromatickej, nemajú jasný klíč podľa ktorého by sme ich perceptuálne zoradili. Preto základné vzťahy *menší ako, väčší ako* nie sú ihned evidentné. Ľudský vizuálny systém vníma vysoké priestorové frekvencie prostredníctvom zmien v jase. Z tohto vyplýva ďalší problém, pretože dúhová paleta zakrýva detail tým, že zmeny medzi hodnotami zobrazuje ako zmenu vo farebnom odtieni a nie ako zmenu v jase. Poslednou, v článku spomenutou, slabinou je, že farby v dúhovej palete sa spájajú do pásem, čím pridáva do obrazu artefakty, ktoré sa nevyskytovali v pôvodných dátach.

Týchto niekoľko závažných dôvodov nás presvedčilo zvoliť inú farebnú paletu. Konkrétnie sme sa rozhodli použiť divergentnú farebnú paletu zloženú z troch farieb: červená, biela a



Obr. 4.9: Vývoj farebnej palety pre farebnú mapu

modrá. Červená znázorňuje kladné maximum, biela hodnoty blízko nuly a modrá záporné maximum. Výber týchto troch farieb bol jednoduchý, pretože červená farba zvyčajne indikuje kladné hodnoty a naopak modrá záporné. Smerom k nule sa potom už iba zvyšuje jasová zložka farby, až kým nedosiahneme bielu. Konkrétnie farby sme vybrali z interneto-vej galérie farebných paliet, ktorú zverejnili *NCAR* (http://www.ncl.ucar.edu/Document/Graphics/color_table_gallery.shtml).

Tento typ palety už lepšie zodpovedal našim požiadavkám. Testovaním sme však zistili, že nie je jednoduché ihneď rozoznať záporné oproti kladným hodnotám, ktoré sú blízko nuly a naopak. Vyriešili sme to nahradením bielej farby dvoma rôznymi farbami. Jednak svetlo modrou pre hodnoty blížiace sa k nule z dola a svetlo červenou pre hodnoty blížiace sa k nule zhora. Vďaka tomu na farebnej škále a rovnako aj vo vizualizácii vznikol ostrý predel medzi kladnými a zápornými hodnotami. Na obrázku 4.9 môžme vidieť porovnanie týchto troch farebných paliet.

4.6 Návrh rozloženia prvkov vizualizácie

Výsledná obrazovka systému sa nebude skladať iba z jedného typu vizualizácie, ale z viacerých grafov a iných častí, ktoré spolu užívateľovi vytvoria dostatočne dobrú predstavu o dátach. Tieto časti budeme nazývať prvky alebo komponenty vizualizácie. Jednotlivým prvkom sme dali nasledovné pomenovania:

- *Meta informácie* je tabuľka obsahujúca informácie o stanici (Názov, geografická poloha, nadmorská výška), názov modelu, interval verifikácie, spôsob interpolácie a podobne.
- *Prehľad* je komponenta vizualizácie, ktorý by mal poskytovať celistvý pohľad na

všetky mesačné štatistiky. Jeho úlohou nie je detailné zobrazenie chýb, ale odhadenie vzorov, trendov a výkyvov vo fungovaní modelu počas skúmaného intervalu.

- *Zjednodušený prehľad* je vo svojej podstate rovnaký ako *Prehľad*. Rozdielom je však menšia plocha, ktorú zaberá, zjednodušená farebná škála a chýbajúce popisky.
- *Celkový priebeh* zobrazuje priebeh chýb počas celého intervalu. Nie však z pohľadu 48 predpovedných hodín (ako *Prehľad*), ale z pohľadu jednotlivých dní.
- *Detail štatistik* je prvok slúžiaci na detailnejšie skúmanie štatistik počas jedného mesiaca. V grafe sa nachádzajú všetky dostupné štatistiky pre daný mesiac.
- *Detail chýb* pre daný mesiac, nám dáva pohľad na chyby predpovede, z ktorých sa počítali štatistiky.
- *Distribúcia chýb* sa taktiež zobrazuje pre konkrétny mesiac a týmto komponentom je graf, v ktorom je zobrazená distribúcia chýb v danom mesiaci.

Pre lepšiu orientáciu v kapitole sme pre jednotlivé prvky priradili aj obrázkové ikony (pozri obrázok 4.10).

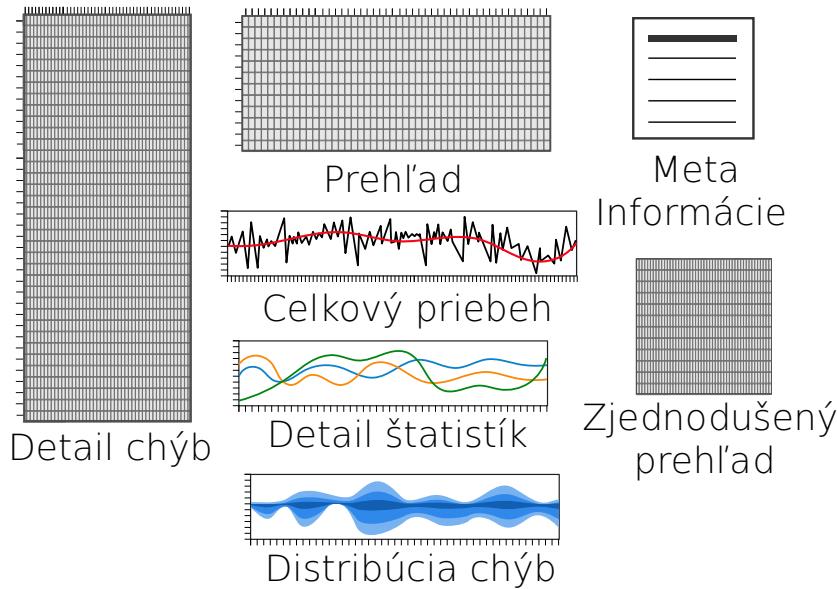
Pri návrhu rozloženie prvkov vizualizácie na obrazovke sme sa rozhodovali medzi dvoma smermi uvažovania, ktoré opíšeme v tejto sekcií.

4.6.1 Viacúrovňový návrh rozloženia prvkov

Jeden zo spôsobov uvažovania, ako rozložiť prvky vizualizácie, bolo zobraziť ich v jednotlivých úrovniach detailu:

- *Úroveň rokov*
- *Úroveň mesiacov*
- *Úroveň štatistik*
- *Úroveň chýb*

Každá úroveň sa skladá z jedného alebo viacerých prvkov rovnakého typu. Napríklad *Úroveň rokov* obsahuje iba prvky *Zjednodušený prehľad*, ktorými sa zobrazujú celé roky.



Obr. 4.10: Ikony pre jednotlivé prvky vizualizácie

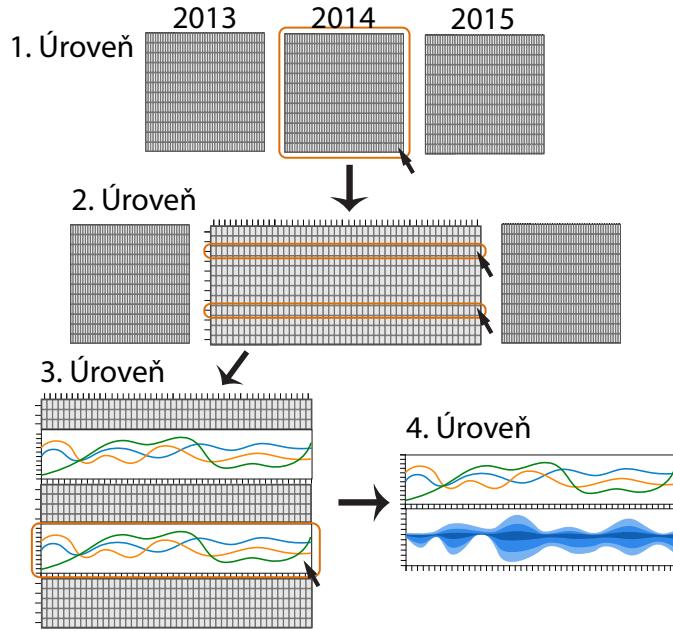
Obmedzením, ktoré kladieme na tento návrh, je zobrazenie maximálne dvoch úrovní súčasne. Touto požiadavkou dosiahneme jednak to, že prvky nebudú zbytočne presahovať mimo obrazovku a taktiež sa zachová do istej miery *detail a kontext*.

Medzi jednotlivými úrovňami je možný pohyb pomocou klikaním myši na jednotlivé prvky. Napríklad v úrovni rokov kliknutím na jeden rok (prvok *Zjednodušený prehľad*) sa nám zobrazí ďalšia úroveň ako prvok *Prehľad*. Ďalším kliknutím na jeden mesiac v danom prvku sa pre daný mesiac zobrazí úroveň štatistik, ako prvok *Detail štatistik*. Ďalej po kliknutí na jednu zo štatistik sa zobrazí prvok *Distribúcia chýb*, v ktorom máme detailne zobrazené chyby, z ktorých sa nám daná štatistika počítala.

Na obrázku 4.11 máme pomocou ikon z obrázka 4.10 schematicky popísané jednotlivé úrovne a pohyb medzi nimi. Môžme si všimnúť, že v schéme nie je vidno prvok *Detail chýb*. Dôvodom je, že sa nám kvôli jeho veľkosti nepodarilo nájsť umiestnenie tohto prvku jednak medzi úrovňami a rovnako aj v rámci úrovne.

4.6.2 Plochý návrh rozloženia prvkov

Pre užívateľa nie je príjemné, keď sa na obrazovke prvky zväčšujú, zmenšujú, posúvajú, objavujú alebo miznú, pretože pri každej modifikácii obrazovky sa užívateľ musí zozna-



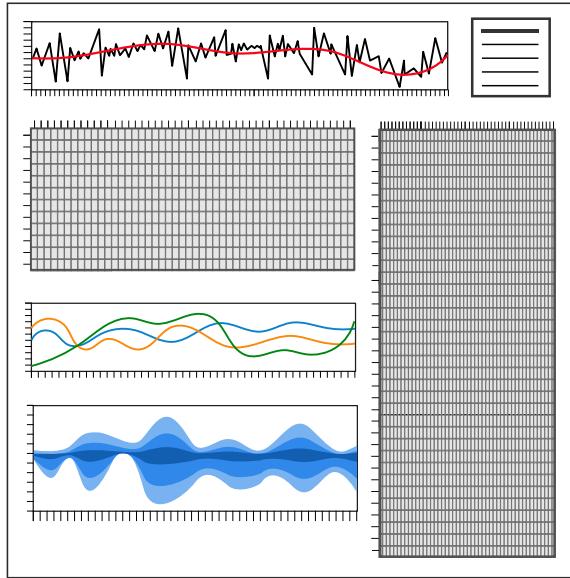
Obr. 4.11: Schematické zobrazenie viacúrovňového návrhu rozloženia prvkov vizualizácie

movať s obrazovkou znova. Preto sme navrhli iný spôsob rozloženia, ktorý tento problém rieši tak, že zobrazí všetky prvky súčasne na jednej úrovni, a teda pojem *úroveň* už nie je potrebný. Takýto návrh rozloženia nie je ničím novým a vo vizualizačnej praxi má názov *multiple views*, teda *viacero pohľadov* na dátá súčasne. Wang Baldonado et al. vo svojej práci navrhol niekoľko pokynov, ktorých sa držať pri dobrom návrhu takéhoto prístupu [WBWK00].

Spôsob, akým sme umiestnili jednotlivé prvky je schematicky znázornený na obrázku 4.12. Ak potrebujeme vidieť detailnejšie informácie o výkone modelu v mesiaci, tak kliknutím na konkrétny mesiac v prvku *Prehľad*, sa zobrazí *Detail štatistik*, *Distribúcia chýb*, *Detail chýb* pre daný mesiac.

Nevýhodou tohto návrhu je, že nemožno zobraziť viacero prvkov jedného typu súčasne. Z tohto dôvodu nie je možné porovnávať viacero rokov súčasne a preto nie je potrebný komponent *Zjednodušený prehľad*. Zvyšné potrebné porovnávania pre mesiace a distribúciu zaobstará v zníženom, ale dostatočnom detaile, prvok *Prehľad*.

Pri návrhu rozmiestnenia prvkov vizualizácie sme chápali komponent *Prehľad* ako hlavnú časť obrazovky vizualizácie, preto sme sa snažili ostatné prvky rozmiestniť odvíjajúc sa od jeho polohy. V prvom rade sme 2 komponenty s rovnakou škálou ako *Prehľad* (*Detail*



Obr. 4.12: Schematické zobrazenie plochého návrhu rozloženia prvkov vizualizácie

štatistik, Distribúcia chýb) umiestnili pod neho, tak aby jednotlivé škály boli na seba zarovnané. Smerom nadol teda stúpa úroveň detailu skúmaných dát. Rovnako sme uvažovali aj pri umiestňovaní prvku Celkový priebeh do hornej časti obrazovky, keďže ho chápeme ako prvok s najnižším detailom. Tento prvok sme sa taktiež snažili rozšíriť čo najviac, keďže škála na x -ovej osi má zvyčajne až 365 hodnôt. Na záver sme umiestnili Detail chýb v pravej časti popri ostatným prvkom, kvôli jeho vertikálne predĺženom tvare.

Kapitola 5

Návrh systému a Implementácia

V tejto kapitole sa venujeme stručnému opisu návrhu systému a jeho jednotlivých častí a taktiež opisujeme kľúčové prvky implementácie.

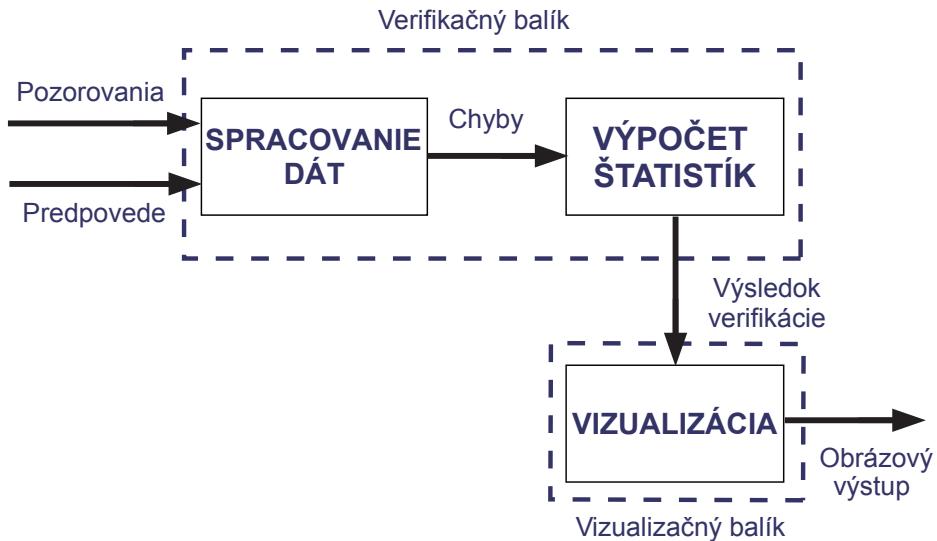
Pri návrhu a implementácii našej aplikácie sme brali do úvahy výhody a obmedzenia systému *IMS4* (*Integrated Monitoring System*), ktorý je vyvíjaný spoločnosťou *MicroStep-MIS s.r.o.*. Dôvodom je, že náš verifikačný systém bol navrhovaný ako prídavný modul pre IMS4.

5.1 Návrh systému

Našu aplikáciu sme sa snažili navrhnúť jednak ako samostatný produkt schopný extrakcie a spracovania meteorologických dát a výpočtu verifikačných štatistik, ale taktiež aj ako súčasť systému IMS4. Toto sme dosiahli tak, že sme rozdelili systém na 2 balíky: *verifikačný* a *vizualizačný*. Na obrázku 5.1 môžme vidieť schematický návrh systému.

Verifikačný balík tvorí jadro celého systému a zohráva viacero úloh. V prvom rade je jeho úlohou extrakcia dát z rôznych dátových zdrojov do tabuľiek predpovedí a pozorovaní. Tieto tabuľky sa následne predspracúvajú (párovanie, konverzia fyzikálnych jednotiek, filtrovanie, identifikácia chýbajúcich dát...) a posúvajú sa do časti na výpočet štatistik. Na základe konfigurácie sa spočítajú rôzne štatistiky, ktorých výsledok je výstup z tohto balíka.

Vizualizačný balík chápeme ako vymeniteľnú časť systému. Vďaka tomuto môže byť



Obr. 5.1: Schematický popis systému.

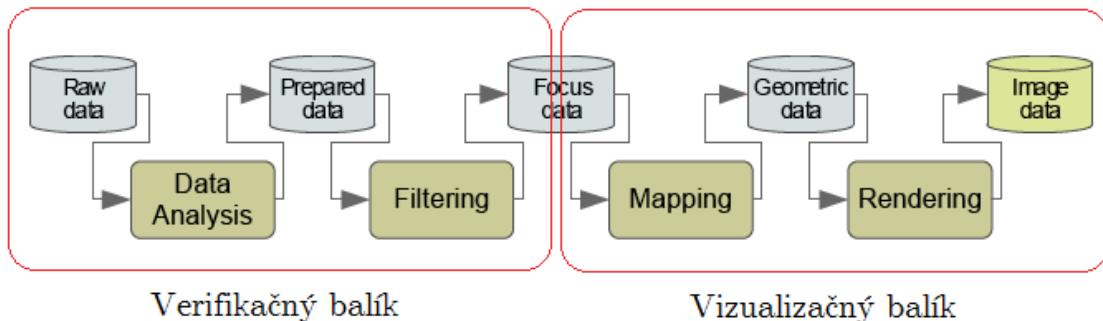
vizualizácia v podobe interaktívnej obrazovky, ktorá je súčasťou systému IMS4, ale taktiež generovaná automaticky do statických obrázkov na disku.

Ak by sme mali systém opísať v pojoch vizualizácie informácií, tak tieto dva balíky rozdeľujú vizualizačnú *pipeline* na dve časti (Pozri obrázok 5.2). Verifikačný balík sa stará o analýzu dát pomocou predspracovania a matematického modelu verifikačných štatistik a taktiež sa stará o filtrovanie dát. Zvyčajne filtrovanie prebieha na základe interakcie užívateľa. V našom prípade sme však uvažovali aj o možnosti, že vizualizácia nebude interaktívna, preto sme filtrovanie umiestnili do verifikačného balíka a vykonáva sa na základe konfigurácie systému.

Filtrované dáta putujú do vizualizačného balíka. V ňom sa vykonáva zobrazenie hodnôt na vizuálne parametre jednotlivých prvkov vizualizácie, tak ako sme to opísali v kapitole 4 *Návrh vizualizácie*. Takéto dáta sa následne vykreslia do obrázka alebo na obrazovku, čím sa ukončí vizualizačná pipeline.

5.2 Použité technológie

Výber implementačných nástrojov bol podmienený obmedzeniami systému IMS4. Z tohto dôvodu sa návrh jednotlivých častí odvíja čiastočne aj od použitej technológie. Aj preto



Obr. 5.2: Rozdelenie prvkov vizualizačnej pipeline do verifikačného a vizualizačného balíka.
Obrázok pipeline pochádza zo stránky <http://www.infovis-wiki.net>.

uvádzame v našej práci sekciu *Použité technológie* skôr, ako samotný návrh jednotlivých balíkov.

5.2.1 Java

Primárnu časť systému - *verifikačný balík* sme implementovali v jazyku *Java*, konkrétnie vo verzii 1.5. Toto je prvé obmedzenie, ktoré na nás kladie systém IMS4, keďže použitie vyššej verzie (momentálne je dostupná už verzia 1.8) by spôsoboval problém s kompatibilitou.

Java ako programovací jazyk je veľmi rozšírený a oblúbený, vďaka čomu je dostupné pomerne veľké množstvo knižníc, nevynímajúc originálne knižnice zahrnuté v systéme IMS4. Tieto knižnice sme využili pri implementácii aj my. Môžeme spomenúť 3 knižnice, z ktorých sú dve interné a jedna pôvodne externá s pridanou funkcionalistou. Prvé dve spomenuté sú *Log* - používaná na logovanie a *X2O*, ktorá sa používa na mapovanie javaovských objektov na XML štruktúru a späť pomocou *Java Reflection API*. My sme X2O použili pri ukladaní a načítavaní konfiguračných súborov. Posledná knižnica je NetCDF-grib [UCA15b] a pochádza od americkej organizácie *Unidata*, ktorú zastrešuje *UCAR*. Jedná sa o opensource produkt na čítanie dát zo súborov vo formáte GRIB. My využívame upravenú verziu NetCDF-grib-6.0 obzvlášť kvôli Jave 1.5, aj keď sú dostupné novšie verzie¹.

¹Najnovšia je NetCDF-Java-4.5, ktorá zoskupuje všetky ďalšie podobné produkty od Unidata.

5.2.2 JavaScript

Systém IMS4 využíva webové stránky na vytváranie GUI. Z tohto dôvodu sme sa rozhodli, že vizualizáciu budeme produkovať priamo v prehliadači pomocou *JavaScriptu*. Podobne ako Java, aj JavaScript je veľmi populárny a preto vzniklo veľké množstvo rôznych knižníc, medzi ktorými sú aj mnohé určené na vizualizáciu. Medzi najznámejšie z nich patria napríklad: *JavaScript InfoVis Toolkit*, *Highcharts*, *jQuery Visualize*, *JS Charts*, *jqPlot*, *jpGraph*, *Raphaël*, *Dygraphs*, *Processing.js*, *Axiis*, *D3* a mnohé ďalšie. Pre naše účely sme z veľkého množstva knižníc vybrali práve D3, ako najvhodnejší nástroj na nás účel.

D3

D3 (*Data-Driven Documents*) [Bos13] je opensource knižnica napísaná v *JavaScripte*. Na vizualizáciu využíva HTML a SVG elementy a ich vizuálne vlastnosti, ktoré mení pomocou ich atribútov a CSS štýlov.

Dôvodom, prečo sme si vybrali práve D3 z veľkého množstva vizualizačných knižníc je, že väčšina z nich bola zameraná na konkrétny druh vizualizácie (napr. *force-directed layout*) alebo na niekoľko najpoužívanejších druhov diagramov (bodový, stĺpcový, čiarový, histogram,...). D3 sa však nezameriava na konkrétny druh vizualizácie, ale ponúka spôsob, ako zobrazovať hodnoty na vizuálne parametre elementov, čím umožňuje ohromnú variabilitu a je vhodná na vytváranie nových alebo menej používaných typov vizualizácií. Väčšina vizualizácií, ktoré sme v našej práci navrhli, nie sú podporované vyššie spomenutými knižnicami a preto sme potrebovali nástroj umožňujúci dobre parametrizovateľnú vizualizáciu.

Toto je taktiež dôvod, prečo vzniklo veľké množstvo knižníc, ukážkových príkladov a návodov ktoré využívajú práve knižnicu D3. Aby sme spomenuli aspoň niektoré z knižníc používajúcich D3, tak sú to napríklad *Raw*, *Cubism*, *Ember Charts*, *NVD3*, *C3*, *MetricsGraphics*, *Graffine*, *Mermaid*, *Epoch*, *Insights*, *Dashku*, *RickShaw* a mnohé ďalšie.

5.3 Verifikačný balík

Ako sme už spomenuli v sekcii 5.1 , tento balík sa skladá z dvoch logických častí: *Spracovanie dát* a *Výpočet štatistik*. V tejto sekcii sa pokúsime zhrnúť kľúčové prvky návrhu a implementácie týchto dvoch častí.

5.3.1 Spracovanie dát

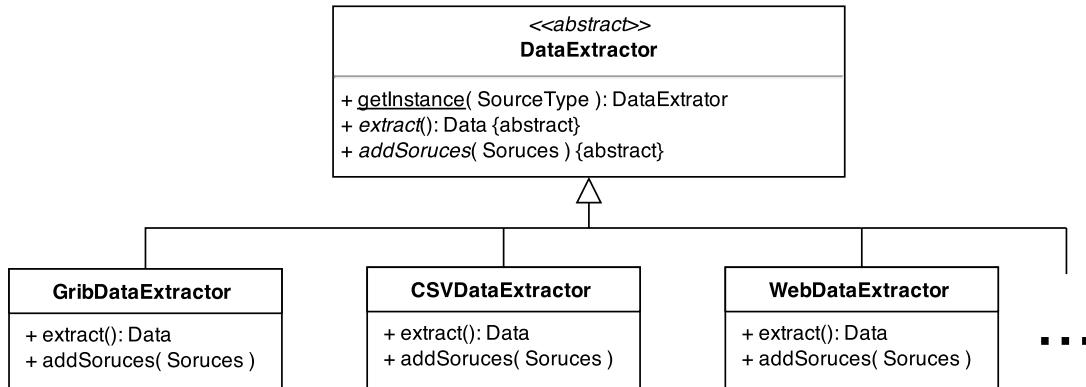
Úlohou tejto časti je získanie potrebných dát z určených dátových zdrojov a ich predspracovanie tak, aby mohli vstúpiť do ďalšej časti - *Výpočet štatistik*.

Pri návrhu tried na získavanie predpovedí a pozorovaní sme sa snažili, aby bolo užívateľovi umožnené získavať dátá z viacerých typov zdrojov: Gribovské súbory, CSV súbory, webové zdroje, databáza a podobne. Aby sme pre každý typ zdroju nevytvárali samostatnú vetvu udalostí, navrhli sme abstraktnú triedu *DataExtractor*, ktorá obsahuje abstraktné metódy *extract()* a *addSources()* a taktiež slúži na inštancovanie všetkých jej potomkov (pozri UML diagram na obrázku 5.3). Jednotlivý potomkovia následne musia implementovať vyššie spomenuté metódy, ktoré sú špecifické pre každý typ zdroju. Napríklad pre webové zdroje metóda *addSources()* očakáva url adresu, zatiaľ čo pre dátové súbory očakáva cestu.

Z extrakcie dát získame tabuľky hodnôt s pozorovaniami a predpovedami pre konkrétné časy. Tieto dáta sa následné predspracúvajú v štyroch krokoch:

1. *Párovanie* : Prebieha tak ako je opísané v podsekcii 1.2.3.
2. *Filtrovanie* : Na základe konfigurácie získavame iba dátumy zvolené užívateľom.
3. *Konverzia jednotiek* : Fyzikálne jednotky z predpovedí a pozorovaní konvertujeme do rovnakej jednotky podľa konfigurácie, aby bol výpočet chýb korektný.
4. *Výpočet chýb* : Prebieha tak ako je opísané v sekciu 1.4.

Tak ako je naznačené na obrázku 5.1, takto spracované dáta putujú do časti výpočet štatistik.

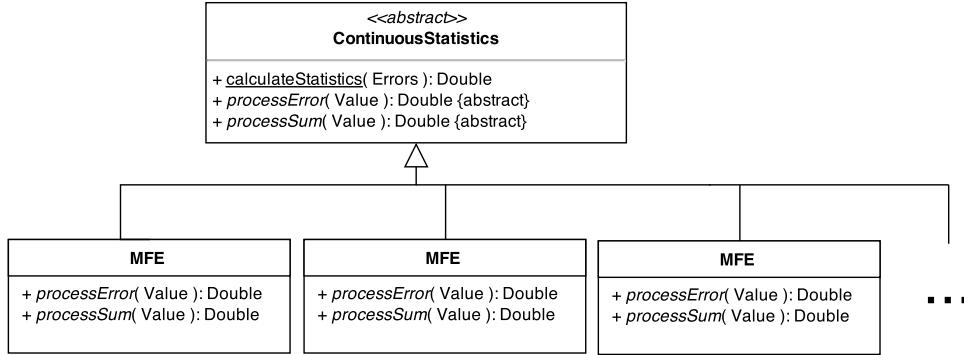


Obr. 5.3: Triedny UML diagram pre triedy typu DataExtractor.

5.3.2 Výpočet štatistik

V tejto časti verifikačného balíka sa vykonáva výpočet štatistik, ktoré sme opísali v sekcií 1.4. V nej sme taktiež navrhli vzorec, ktorý opisuje výpočet kumulovanej chyby vo všeobecnosti pomocou dvoch funkcií ε a Φ (pozri podsekcii 1.4.4).

Pre pripomienanie, funkcie ε a Φ sú ľubovoľnými funkiami z \mathbb{R} do \mathbb{R} a všeobecný vzorec pre kumulované skóre je nasledovný: $\Phi\left(\sum_{i=0}^n \varepsilon(e_i)\right)$. Ako vidíme ε slúži na spracovávanie chyby a Φ na spracovávanie sumy. Toto riešenie sme použili aj pri návrhu a implementácii programových tried na výpočet štatistik. Na obrázku 5.4 môžeme v triednom diagrame vidieť, že hlavným telom celého návrhu je abstraktná trieda *ContinuousStatistics* obsahujúca dve abstraktné metódy *processError()* a *processSum()*. Práve tieto dve metódy predstavujú funkcie ε a Φ . Trieda *ContinuousStatistics* obsahuje taktiež tretiu - *nie abstraktnú* - metódu *calculateStatistics()*, v ktorej tele sa vykonáva suma všetkých chýb s použitím týchto dvoch metód. Na obrázku 5.4 môžeme taktiež vidieť, že všetky štatistiky sú implementované ako potomkovia triedy *ContinuousStatistics* a teda musia implementovať vyššie spomenuté abstraktné metódy. To ako vyzerajú tieto metódy pre jednotlivé štatistiky sme taktiež ukázali v spomínamej podsekcii 1.4.4.



Obr. 5.4: Triedny UML diagram pre triedy typu ContinuousStatistics.

5.4 Vizualizačný balík

Úlohou vizualizačného balíka je spracovávanie výsledkov verifikácie a vygenerovanie z týchto dát obrazovku vizualizácie navrhnutú v predošlej kapitole.

Pri návrhu a implementácii tohto balíka sme sa snažili využiť niektoré vlastnosti, ktoré nám ponúkajú použité technológie. Veľkú časť tohto balíka sme navrhli ako súčasť knižnice D3 s vyžitím návrhových vzorov charakteristických pre túto knižnicu. Jednak išlo o takzvaný *chaining pattern*, ktorý slúži pri vytváraní objektu. Tento pattern dovoľuje volanie metód objektu, ktorými sa nastavujú jeho vlastnosti, zretežených za sebou. Príklad pre lepšie pochopenie je ukázaný na výpise 5.1. Ďalším vzorom je *callback pattern*, kedy sa používa funkcia ako parameter.

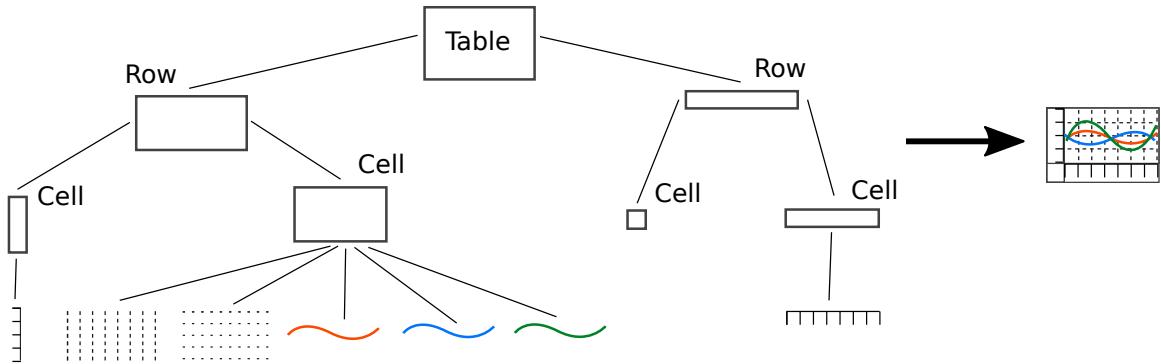
```

1 var groups = group.selectAll(".group")
2     .data(data)
3     .enter()
4     .append("svg:g")
5     .classed("group", true);

```

Výpis 5.1: Ukážka aplikácie návrhového vzoru *chaining*.

Jednotlivé prvky vizualizácie sme navrhli ako takzvané *moduly*, ktoré obsahujú v sebe dátá, nastavenia zobrazenia a ďalšie vnorené moduly, ako svojich potomkov. Každý modul je funkcia, ktorá ma na vstupe *selection*, teda DOM element vybraný pomocou knižnice D3. Tento typ funkcie je *callback* funkcia, ktorú jednotlivé vybrané elementy knižnice D3 volajú pomocou metódy `call()`.



Obr. 5.5: Príklad štruktúry modulov pre vytvorenie mnoho-čiarového diagramu.

Tento návrh nám vytvára akúsi stromovú štruktúru modulov (pozri obrázok 5.5), ktorá umožňuje jednoduché vystavovanie celého rozloženia prvkov, kombinovanie jednotlivých častí vizualizácie a znovupoužitie niektorých častí vizualizácie.

Príkladom znovupoužitia môže byť graf pre distribúciu chýb, ktorý sa skladá z horizontálnej a vertikálnej osi a samotnej vizualizácie, ktorá môže byť rôzna (pozri sekciu 4.4). Pre rôzne typy vizualizácií teda nemusíme kopírovať rovnaký kus kódu na generovanie osí viackrát, ale iba recyklujeme opakujúcu sa časť grafu.

Každý typ grafu sme taktiež rozobrali na základné časti, z ktorých sa skladá. Napríklad mnoho-čiarový diagram sa skladá z osí, horizontálnej a vertikálnej mriežky a mnohých čiar. Vďaka tomuto jednako môžeme jednoducho riadiť počet čiar alebo mriežok, ale taktiež môžeme čiarový diagram kombinovať so stĺpcovým, s bodovým alebo ľubovoľným iným diagramom. Zjednodušene povedané, tým, že každý element chápeme ako samostatný modul, získavame dobrú variabilitu vizualizácie.

Kapitola 6

Výsledky

6.1 Testovanie

Pri testovaní vizualizácie sme postupovali podľa článku od Elmquist et al. [EY12], ktorý opisuje najpoužívanejšie vzory testovania vizualizácie a článok od Vogel et al., v ktorom autori testujú webovú vizualizáciu [V ро KМK11].

6.1.1 Testovacia procedúra

Testovanie prebiehalo v dvoch častiach prostredníctvom niekoľkých úloh, ktoré sme navrhli na základe špecifikácie požiadaviek na vizualizáciu uvedených v sekcii 4.2.

V prvej časti sme testovali náš systém a v časti druhej starý používaný spôsob po-zostávajúci z čiarových diagramov pre jednotlivé mesiace v roku (pozri prílohu A.2).

V oboch časti bola každému testovaciemu subjektu vysvetlená podstata verifikácie a predstavená základná funkcionálita systému v krátkom 15 minútovom návode. Následne subjekt obdržal testovací formulár (pozri prílohu A.3) obsahujúci 7 rôznych úloh, ktoré mal subjekt vykonať a po vykonaní každej z nich do formulára zapísat výsledok svojho skúmania.

Jednotlivé časti testovania boli vykonané následne za sebou na rovnakých testovacích subjektoch. Teda po testovaní prvého systému rovnaký subjekt testoval druhý systém.

Pri testovaní sme jednak skúmali správnosť odpovedí ale aj rýchlosť vykonania jednotlivých úloh.

6.1.2 Testovacie dátá

Na testovanie sme použili reálne dátá predpovedí modelu **WRF** pokrývajúce celý Arabský poloostrov a pozorovania (merania) z konkrétnych meteorologických staníc v meste Dubaj.

Ked'že sme robili porovnávanie testovanie dvoch systémov s rovnakou množinou užívateľov, rozhodli sme sa pre rôzne systémy použiť rôzne dátá. Takýmto spôsobom sme sa snažili vylúčiť to, že subjekt použije už naučené poznatky z predošlého pokusu, ked'že našou snahou bolo získať, čo najpravdivejšie výsledky testovania.

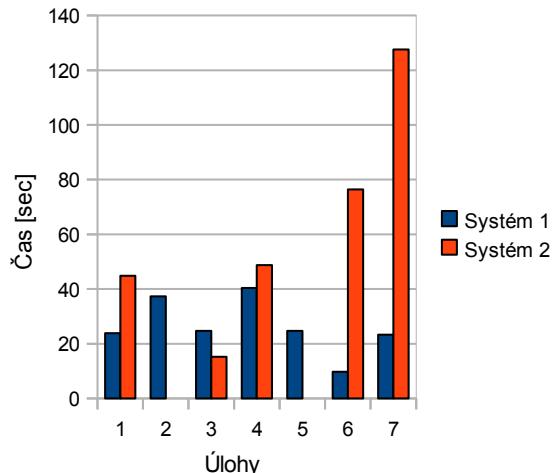
Náš systém sme testovali na predpovediach *tlaku* z roku 2012 pre Dubajskú stanicu *Al Faqa*, zatiaľ čo starý systém sme testovali na predpovediach *teploty* z roku 2013 pre stanicu *Jebel Ali*. Použili sme teda dátá pre rozdielnú veličinu, rok a stanicu.

6.1.3 Výsledky testovania

Vizualizáciu sme testovali na 6 subjektoch bez vyššieho meteorologického, matematického alebo informatického vzdelania. Taktiež žiadnen užívateľ nemal skúsenosti s verifikáciou predpovedí a náš systém videl prvýkrát v živote. Všetky tieto faktory vplývali na výsledok nášho testu, ked'že sme sa často stretali s nepochopením respektíve s pomalým pochopením zadania a teda čas riešenia úlohy sa značne natiahol.

Tabuľka 6.1: Výsledky testovania

Úloha	Systém 1	Systém 2	Priemerný čas 1	Priemerný čas 2
1	6/6	3/6	23.85s	44.81s
2	6/6	—	37.27s	—
3	6/6	4/6	24.66s	15.2s
4	6/6	2/6	40.36s	48.69s
5	6/6	—	24.69s	—
6	6/6	4/6	9.73s	73.37s
7	6/6	1/6	23.27s	127.53s
Suma	42/42	14/30	183.83s	312.5s



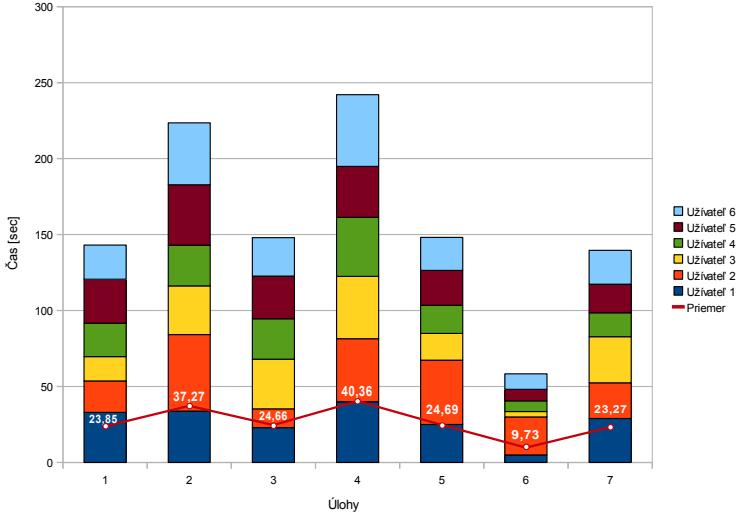
Obr. 6.1: Porovnanie časov nášho (Systém 1) a starého systému (Systém 2)

V tabuľke 6.1 vidíme, že v našom systéme (Systém 1) boli všetci užívatelia stopercenné úspešný a teda všetky úlohy vyriešili správne. Avšak v starom systéme (Systém 2) bola úspešnosť len polovičná, pričom dve úlohy nebolo možné vyriešiť pomocou danej vizualizácie vôbec.

Rovnako aj v nameraných časoch bol náš systém výrazne lepší. Celkový čas potrebný na riešenie 7 úloh zabral testovaným užívateľom v priemere 3 minúty, a vykonanie jednej zabralo v priemere asi pol minúty (26s), zatiaľ čo v starom systéme vyriešenie 5 úloh trvalo vyše 5 minút a každá úloha zabrala v priemere 1 minútu (62s).

V tabuľke sme tiež zvýraznili najdlhší a najkratší priemerný čas z vykonávaných úloh. Vidíme, že úloha, ktorá bola v našom systéme očividne najjednoduchšie riešiteľná mala v starom systéme druhý najhorší priemerný čas. Avšak úloha, ktorá zabrala v našom systéme najviac času trvala podobný (aj keď o niečo lepší) čas ako pre starý systém, nehladiac na to, že riešenie úlohy v starom systéme bolo pomerne neúspešné a to iba 2 z 6 subjektov odpovedalo správne. Lepšie porovnanie jednotlivých časov pre dané úlohy môžeme vidieť v grafe na obrázku 6.1.

Ak by sme mali zhodnotiť náš systém v porovnaní so starým na základe výsledkov testovania, môžme povedať, že riešenie jednotlivých úloh analýzy dát je v našom systéme pomocou nami navrhнутej vizualizácie výrazne jednoduchšie a presnejšie.



Obr. 6.2: Výsledky testovania nášho systému pre jednotlivých užívateľov.

Zhodnotenie výsledkov nášho systému

V grafe na obrázku 6.2 môžme lepšie vidieť sumárne porovnanie časov pre jednotlivé úlohy a užívateľov. Môžme si ľahko všimnúť, že úlohy 1, 2, 5 a 7 majú približne rovnaký priemerný čas, zatiaľ čo úlohy 2, 4 mali výrazne dlhšie trvanie a úloha 9 trvala v priemere najkratšie.

Dlhé trvanie úlohy 2 prisudzujeme vyššej zložitosti úlohy oproti ostatným. Pre užívateľov bolo odhalenie outlierov pomerne rýchle, avšak väčšinu času zabralo užívateľom odčítanie z grafu o akú presnú hodnotu a predpoved' ide.

Pri úlohe 4 bolo problémom rýchle odčítanie z grafu, o aký dátum sa jedná. Dôvodom bolo nedobré označenie dátumov na x -ovej škále, kedy sa jednotlivé dni v roku označujú číslami 0-364. Užívateľ uvidel konkrétny dátum, až keď nadišiel myšou na dané miesto, kedy sa mu presné hodnoty ukázali v *tooltip*. Lepším riešením by bola redšia škála s mesačnými intervalmi alebo konkrétnymi dátumami.

Výsledky nášho testovania považujeme celkovo za uspokojivé, keďže aj neskúsení užívatelia dosahovali pri všetkých úlohách dobré časy. Taktiež nám tieto výsledky poukázali na slabiny našej práce, ktorými sú hlavne zlé alebo slabé označenia škál, grafov a legiend.

6.2 Demonštrácia

V tejto časti chceme demonštrovať príklad analýzy fungovania predpovedného modelu pomocou nami navrhnutých techník vizualizácie. Rovnako ako pri testovaní vizualizácie aj tu sme použili predpovedané dátá z modelu WRF. Konkrétnie sa jednalo o tlak pri povrchu zeme v hektopaskaloch.

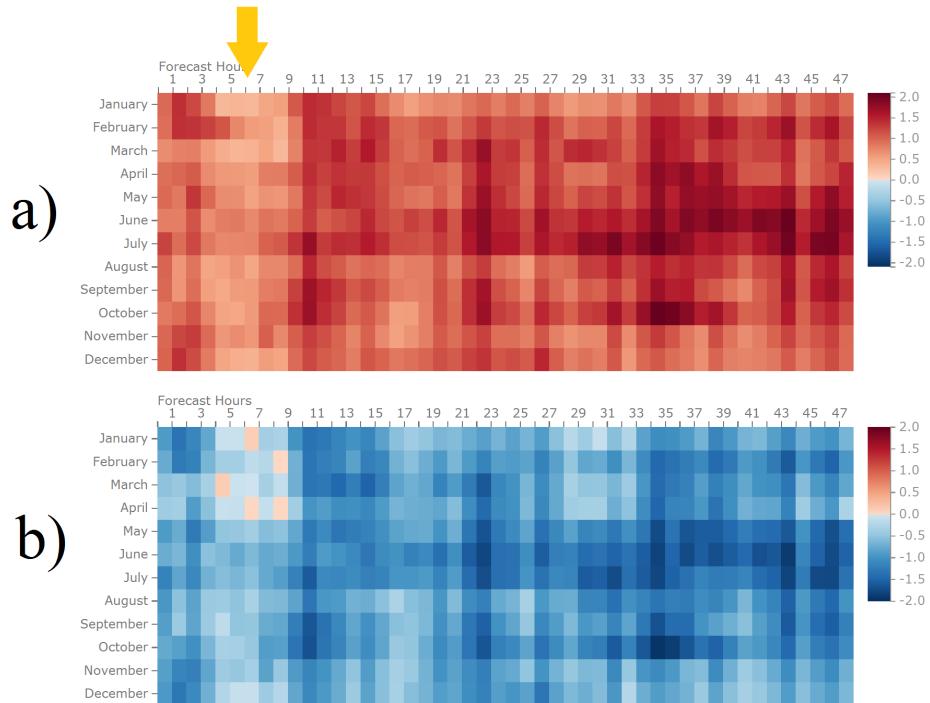
6.2.1 Štatistiky

Na obrázku 6.3a môžme vidieť farebnú mapu (Prehľad) zobrazujúcu RMSE pre celý rok 2012. Ako už vieme zo sekcie 1.4, RMSE nadobúda iba kladné hodnoty a teda nehovorí nič o smeru chyby, ale iba o jej veľkosti.

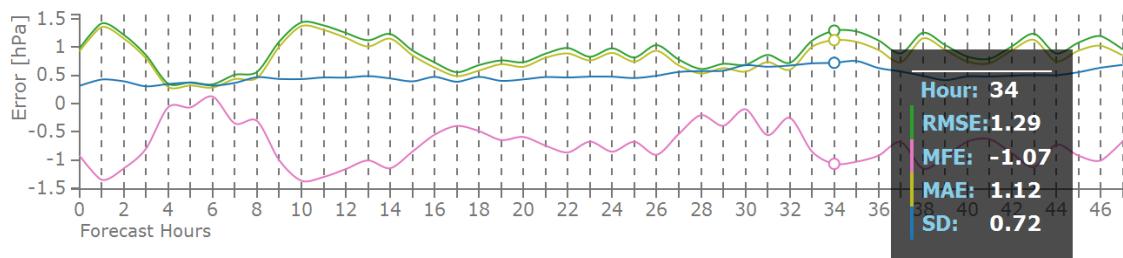
Z daného obrázka ľahko vyčítať niektoré zaujímavé vzory. Ku príkladu je veľmi dobre vidno, že model predpovedá najlepšie v okolí šiestej hodiny predpovede a v okolí poludnia zaznamenávame prudký pokles presnosti predpovedí. Fakt, že model predpovedá najlepšie až okolo šiestej hodiny a nie ihned' na začiatku predpovede trocha nabúra nás predpoklad, že smerom do budúcnosti sa predpoved' zhorsuje. Tu sa potvrdzuje dôležitosť vizualizácie, ktorá slúži nie len na to, aby sme *zistili očakávané*, ale aj *objavili neočakávané*.

Na obrázku 6.3b máme zobrazenú priemernú chybu (MFE), ktorá sa počítala z rovnakých dát ako RMSE na obrázku 6.3a. Na tomto grafe je ihned' vidieť to, či model podhodnocuje alebo nadhodnocuje predpovede. Ako vidíme väčšina grafu je zafarbená na modro, čo značí záporné hodnoty, a teda podhodnotenie predpovedí. Taktiež vďaka dobre zvolenej farebnej škále môžme vidieť, že niektoré hodnoty (napríklad 6 hodina v Januári) sú kladné, avšak stále blízko nuly.

V prípade, že užívateľ potrebuje získať informáciu o veľkosti a aj smere chyby súčasne, po kliknutí na konkrétny mesiac sa mu zobrazia všetky vypočítané štatistiky súčasne (pozri obrázok 6.4). O aké štatistika sa jedná vidíme v *tooltip* podľa farebného rozlíšenia.



Obr. 6.3: a) Farebná mapa zobrazujúca RMSE b) Farebná mapa zobrazujúca MFE



Obr. 6.4: Zobrazenie RMSE, MFE, MAE a SD (Standard Deviation) na jednom grafe súčasne

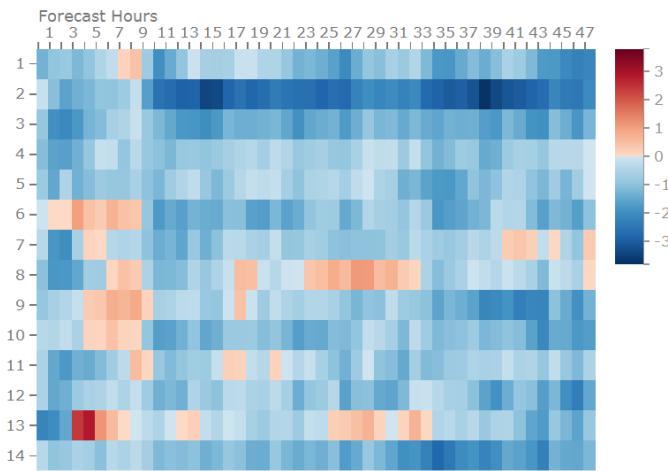
6.2.2 Chyby

Súčasťou obrazovky vizualizácie je aj zobrazenie jednotlivých chýb, z ktorých sa počítali štatistiky, či už pomocou farebnej mapy alebo pomocou grafov pre vizualizáciu distribúcie chyby.

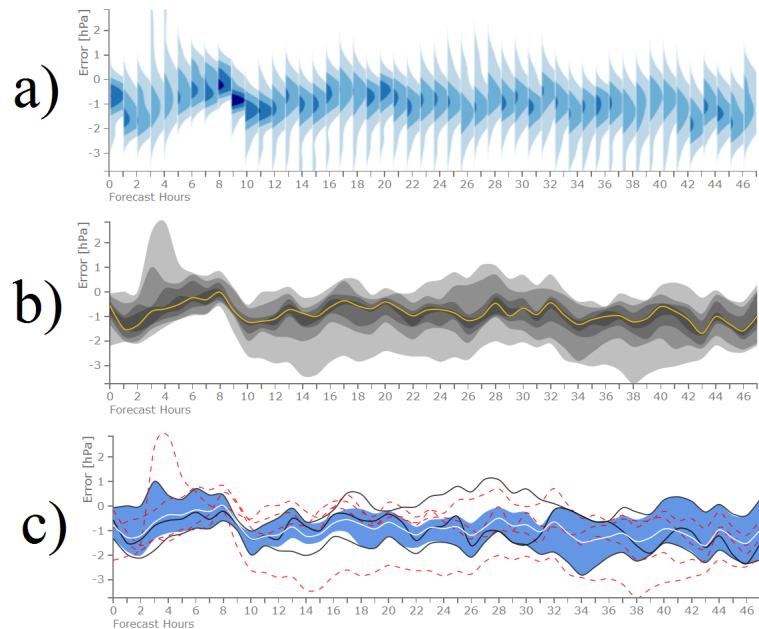
Farebná mapa na obrázku 6.5 nám dáva dobrý obraz o chybách v chronologicky zo- radených predpovediach pre mesiac Február. Na tomto grafe si môžme všimnúť niektoré predpovede, ktoré boli celkom úspešné - napríklad predpovede 4,5 a 12 majú chyby pomerne blízko 0, ale aj predpovede, ktoré sa nejakým spôsobom odlišujú od ostatných. Napríklad väčšina predpovedí podhodnocuje v hodinách 23-31, zatiaľ čo predpoved' 8 a 13 nadhodnocuje. Ďalej vidíme, že predpoved' 13 v okolí štvrtnej hodiny má výrazný výkyv hodnôt a predpoved' 2 celkovo výrazne podhodnocuje.

Tento typ vizualizácie odporúčame kombinovať s jedným z grafom distribúcie, ktoré sú na obrázku 6.6. Konkrétnie na funkčnom krabicovom diagrame (obrázok 6.6c) je veľmi dobre vidieť tri predpovede, znázornené červenou prerusovanou čiarou, ktoré sa výrazne líšia od ostatných. Sú to už spomínané predpovede 2, 8 a 13. Výrazný oblúk v okolí tretej hodiny patrí výkyvu hodnôt pre predpoved' 13, ktorý je ľahký objaviť aj v grafe na obrázku 6.5. Okrem tohto výkyvu nám graf ukazuje, že predpoved' 13 sa zvyšnou časťou krivky nachádza v centrálnej oblasti grafu, teda jej úspešnosť pre iné neskoršie hodiny je podobná ostatným predpovediam.

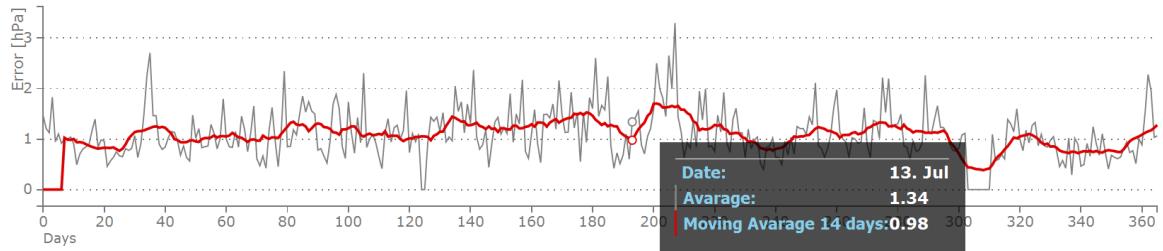
Grafy na obrázkoch 6.6a a 6.6b nám podávajú rovnakú informáciu, avšak v rozdielnom detaile. Na oboch grafoch vidíme, že v okolí deviatej predpovednej hodiny majú chyby najmenší rozptyl. Vidíme aj, že v okolí štvrtnej hodiny je naopak rozptyl väčší, avšak najväčšia hustota chýb sa stále nachádza blízko priemernej hodnoty, teda napriek výkyvom sú štatistiky z týchto dát pomerne dôveryhodné. Z grafu na obrázku 6.6a možno vyčítať, že chyby pre niektoré hodiny majú už takmer bimodálnu distribúciu - druhá a posledná hodina predpovede.



Obr. 6.5: Farebná mapa chýb predpovedí pre mesiac Február. Na y -ovej osi je poradie predpovede.



Obr. 6.6: Grafy distribúcie pre mesiac Február a) Graf hustoty b) Pruhový kvantilový diagram c) Funkčný krabíkový diagram



Obr. 6.7: Graf ročného priebehu chýb predpovedí. Šedou čiarou sú zobrazené denné štatistiky (konkrétnie RMSE) a červenou je kľzavý priemer vypočítaný z týchto štatistik.

6.2.3 Ročný priebeh

Posledným grafom, ktorý sa objavil na našej vizualizačnej obrazovke verifikácie je čiarový graf pre ročný priebeh chýb predpovedí (pozri obrázok 6.7). Na x -ovej osi grafu máme jednotlivé dni v roku (0-364) a na y -ovej veľkosť chyby v hPa. Tento graf zobrazuje denné štatistiky šedou čiarou a červenou je vyhľadená táto krivka pomocou kľzavého priemeru.

Na obrázku 6.7 máme konkrétnie RMSE. Celkovo môžeme vidieť, že v strede roka (v lete) sú chyby najväčšie a niekedy v Novembri sú chyby najnižšie. Dôvodom však môžu byť aj chýbajúce dátá, keďže šedá krivka je podozrivo konštantná rovno na nule. Veľké skoky hodnôt môžeme pozorovať aj niekedy začiatkom Februára respektíve koncom januára a taktiež aj koncom Decembra.

Záver

V našej práci sme uviedli problém verifikácie predpovedných modelov počasia so zameraním na verifikáciu predpovede spojitej premennej v jednom bode. Zhrnuli sme používané štatistické metódy na meranie chyby predpovede a navrhli sme spôsob všeobecného výpočtu kumulovanej chyby, ktorý sme v závere využili aj pri implementácii.

Taktiež sme preskúmali doterajšie riešenia jednak z pohľadu verifikačného softvéru, ale aj z pohľadu vizualizačných techník vo verifikácii. Uviedli sme slabé a silné stránky jednotlivých softvérových riešení a zhrnuli sme ich v prehľadnej tabuľke. Taktiež sme opísali a analyzovali vizualizácie používané vo verifikácii a načrtli ich zameranie a spôsob použitia.

Ďalej sme charakterizovali vstupné dátá a špecifikovali požiadavky používateľov, na základe ktorých sme navrhli rôzne spôsoby vizualizácie, ktoré sme implementovali ako doplnok JavaScriptovej knižnice D3. V závere práce sme následne vykonali testovanie našej vizualizácie na základe čoho sme zhodnotili jej úspešnosť.

Výsledkom práce je teda verifikačný nástroj schopný spracovávať dátá z rôznych zdrojov, ponúkajúc funkciaľitu pre verifikáciu predpovedí spojitej premennej s použitím špeciálne navrhnutej vizualizácie pre tieto účely.

Prínos

Za hlavný prínos tejto práce považujeme práve návrh špecializovanej vizualizácie, ktorá síce nevyniká obrovskou inováciou, ale pri jej vzniku sme navrhli mnoho drobných originálnych vylepšení smerujúcim k lepšiemu, rýchlejšiemu a jednoduchšiemu porozumeniu dát. Príkladom môže byť návrh prehľadovej vizualizácie verifikačných dát, návrh riešenia viacerých škál pri mnoho-čiarovom diagrame, zovšeobecnenie re-dizajnu krabicového dia-

gramu od Bada et. al. [BSM04] na takzvaný *Pruhový diagram*, využitie dvojtónového pseudofarbenia pri kompaktnej vizualizácii distribúcie dát pomocou grafu hustoty a podobne.

Medzi prínosy našej práce patrí aj to, že nami vyvinutý systém a si nájde praktické uplatnenie pri riešení skutočných problémov reálneho sveta. Ku príkladu bol tento systém predaný do Singapúru pre Radmon (*Radiation Monitors*), ale využitie si nájde aj

Budúca práca

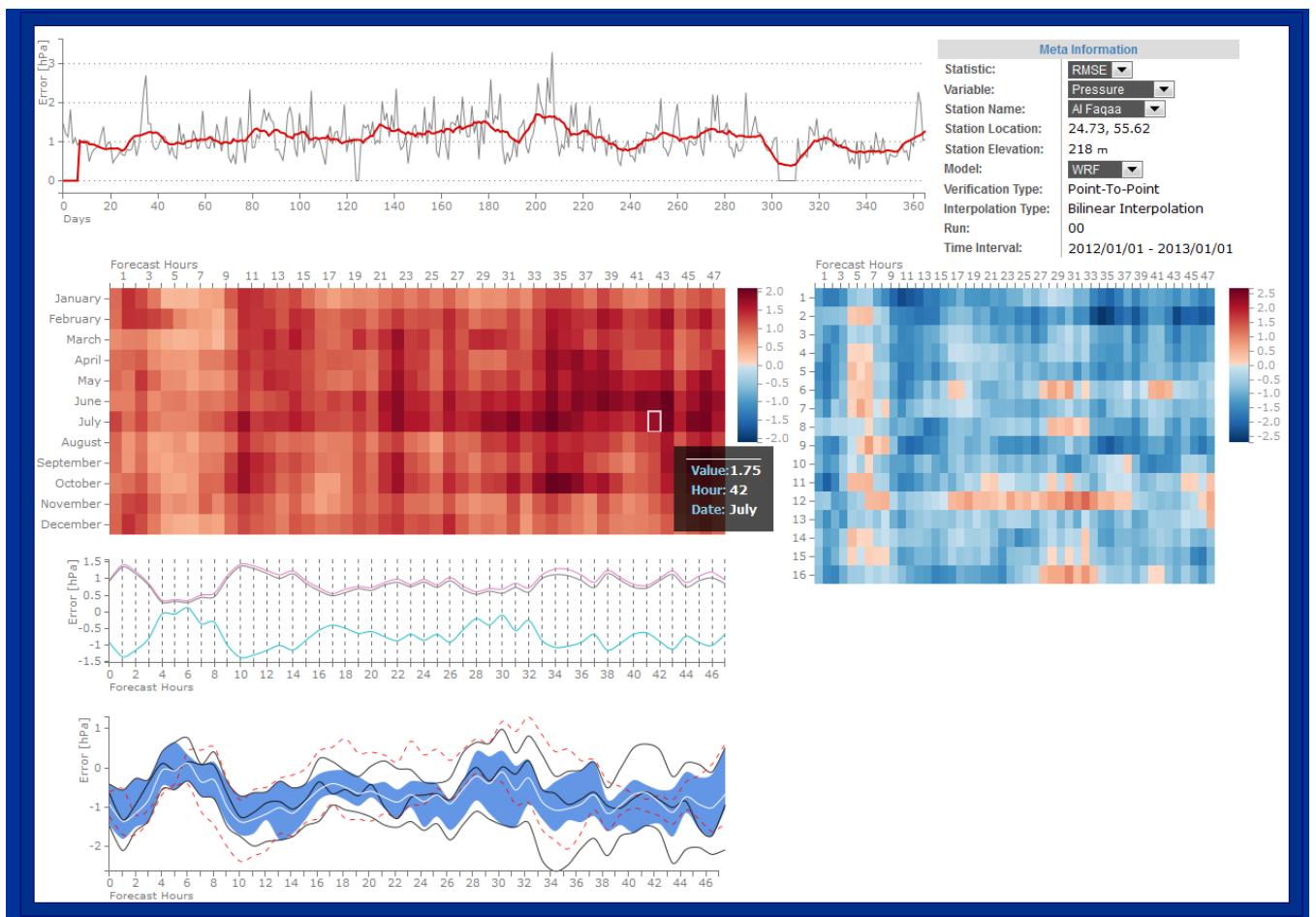
Na výskum a vývoj neustále existuje priestor a v našej práci ho vidíme určite mnogo obzvlášť v štyroch oblastiach:

1. *Získavanie dát* - V práci sme implementovali 3 typy zdrojov: Grib, CSV a Web. Avšak existuje mnogo ďalších zdrojov dát, z kade možno získať pozorovania, rovnako ako aj predpovede. Príkladom je napríklad *databáza*, *XML*, alebo *XLS* súbory, takzvané *bloky* systému IMS4, kde sú aj merania a mnohé iné ďalšie zdroje.
2. *Konfigurácia systému a vizualizácie* - Konfigurovanie systému v momentálnom stave nie je veľmi user-friendly úloha. V budúcej práci očakávame návrh riešenia, ktoré skryje pred užívateľom pomerne neprehľadné nastavenia systému.
3. *Interaktívna manipulácia s dátami* - Momentálne riešenie neponúka veľmi dôležitú súčasť moderných vizualizačných nástrojov a tým je manipulácia s dátami priamo na obrazovke. Príkladmi operácií môže byť filtrovanie dát na základe rôznych kľúčov alebo selekcia vybranej vzorky dát pomocou *brushingu* a podobne.
4. *Porovnávanie modelov a staníc* - Pri návrhu vizualizácie sme nezvažovali možnosť porovnávania výkonu viacerých predpovedných modelov alebo porovnávanie predpovedí pre rôzne stanice súčasne. Riešenie takéhoto problému by pravdepodobne vyžadovalo zásadný re-dizajn vizualizácie.

Dodatok A

Prílohy

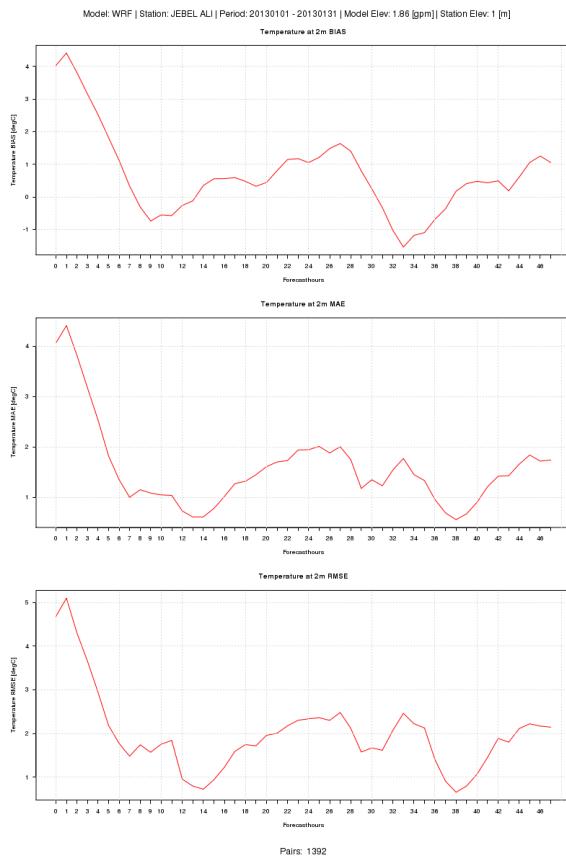
A.1 GUI



Grafické užívateľské rozhranie našej aplikácie pozostáva z jednotlivých prvkov,

tak ako sme to opísali v podsekcii 4.6.2. V pravom hornom rohu si v tabuľke meta informácií môže užívateľ zvoliť zobrazovanú štatistiku, veličinu, stanicu a predpovedný model, ktorý chce verifikovať. Kliknutím na konkrétny mesiac sa užívateľovi zobrazia grafy pre daný mesiac.

A.2 Príklad vizualizácie starého systému



V predošлом systéme verifikovania sa jednotlivé štatistiky vizualizovali pomocou čiarových diagramov a to jeden diagram pre každú štatistiku. Jednotlivé grafy sa generovali pomocou skriptovacieho jazyka R (pozri 2.1.3) do statických obrázkov bez možnosti interakcie.

A.3 Testovací formulár

Testovací formulár

1. Zisti hodnotu RMSE a MFE pre piatu hodinu predpovede (*forecast hour*) v Júni.
RMSE: MFE:
 2. Pre Apríl nájdi 2 hodnoty predpovedí, ktoré sa výrazne líšia od ostatných hodnôt (tzv. outlierov).
 1. Hodnota: Predpoved':
 2. Hodnota: Predpoved':
 3. Nájdi aká predpovedaná hodina (*forecast hour*) mala pre Január najväčšie MFE.
 4. Nájdi, v ktorom mesiaci v roku bola priemerná absolútна chyba (MAE) najnižšia.
 5. Porovnaj rozptyl chýb na začiatku a konci predpovede pre December.
(vyber 1 z troch možností)
 - a) Smerom ku koncu predpovede sa rozptyl výrazne zväčšuje
 - b) Smerom ku koncu predpovede sa rozptyl výrazne zmenšuje
 - c) Počas celej predpovede zostáva rozptyl približne rovnaký
 6. Zisti, či model celoročne skôr nadhodnocuje alebo podhodnocuje predpovede.
(záporné hodnoty = podhodnotenie, kladné hodnoty = nadhodnotenie)
 - a) Nadhodnocuje
 - b) Podhodnocuje
 7. Zisti, ktorú predpovednú hodinu (*forecast hour*) počas celého roka zvyknú byť predpovede najviac a najmenej presné.
(presné = hodnoty štatistik sú blízko 0)
- Najmenej: Najviac:

A.4 CD

Obsah CD:

- Zdrojové súbory
- Testovacie dáta
- Elektronická verzia práce

Literatúra

- [AA05] Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schuman, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction. Springer Verlag, 1st edition, 2011.
- [BA97] Adrian W. Bowman and Adelchi Azzalini. *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations (Oxford Statistical Science Series)*. Oxford University Press, USA, November 1997.
- [BI07] David Borland and Russell M. Taylor II. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17, 2007.
- [BL01] G. J. Boer and S. J. Lambert. Second-order space-time climate difference statistics. *Climate Dynamics*, 17(2):213–218, January 2001.
- [Bos13] Mike Bostock. Data-Driven Documents library. <http://d3js.org/>, 2013. [Prístupné online: 14.4.2015].
- [Bro15] Dr. James D. Brown. *Ensemble Verification Service (EVS) version 5.4 User’s Manual*. National Weather Service’s Office of Hydrologic Development (OHD), Marec 2015.
- [BSM04] Ragnar Bade, Stefan Schlechtweg, and Silvia Miksch. Connecting time-oriented data and information to a coherent interactive visualization.

- In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 105–112, New York, NY, USA, 2004. ACM.
- [Cas09] Barbara Casati. *Verification of continuous predictands*. Joint Working Group on Forecast Verification Research (JWGFVR), Jún 2009.
- [CCM13] Yu-Hsuan Chan, Carlos D. Correa, and Kwan-Liu Ma. The generalized sensitivity scatterplot. *IEEE Trans. Vis. Comput. Graph.*, 19(10):1768–1781, 2013.
- [Cen14] Developmental Testbed Center. *Model Evaluation Tools Version 5.0, User's Guide 5.0*. DTC, September 2014.
- [CWS⁺08] B. Casati, L. J. Wilson, D. B. Stephenson, P. Nurmi, A. Ghelli, M. Pocernich, U. Damrath, E. E. Ebert, B. G. Brown, and S. Mason. Forecast verification: current status and future directions. *Meteorological Applications*, 15(1):3–18, 2008.
- [DTC15] DTC Developmental Testbed Center. Model Evaluation Tools. <http://www.dtcenter.org/met/users/>, Marec 2015. [Prístupné online: 25.3. 2015].
- [EY12] Niklas Elmquist and Ji Soo Yi. Patterns for visualization evaluation. In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, pages 12:1–12:8, New York, NY, USA, 2012. ACM.
- [Few08] Stephen Few. Solutions to the Problem of Over-Plotting in Graphs. *Perceptual Edge*, 2008.
- [FJB12] Tressa L. Fowler, Tara L .Jensen, and Barbara G. Brown. *Introduction to Forecast Verification*. 2012.
- [Fou15] The R Foundation. The R Project for Statistical Computing. <http://www.r-project.org/>, Marec 2015. [Prístupné online: 19.3. 2015].

- [Fri09] Michael Friendly. Milestones in the history of thematic cartography, statistical graphics, and data visualization. 2009. Na webe: <http://datavis.ca/milestones/>.
- [Gol] Professor Brian Golding. Weather forecasting part 1. <http://www.rmets.org/weather-and-climate/weather/weather-forecasting>. [Prístupné online: 6.12.2014].
- [Ins15] SAS Institute. Statistical Analysis Software. <http://www.sas.com>, Marec 2015. [Prístupné online: 19.3. 2015].
- [JHM⁺13] Halldor Janetzko, Ming C. Hao, Sebastian Mittelstädt, Umeshwar Dayal, and Daniel A. Keim. Enhancing scatter plots using ellipsoid pixel placement and shading. In *46th Hawaii International Conference on System Sciences, HICSS 2013, Wailea, HI, USA, January 7-10, 2013*, pages 1522–1531, 2013.
- [JME10] Waqas Javed, Bryan McDonnel, and Niklas Elmquist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, November 2010.
- [Lab14] NCAR Research Applications Laboratory. *Weather Forecast Verification Utilities*. NCAR, Júl 2014.
- [LPR09] Sara López-Pintado and Juan Romo. On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718–734, 2009.
- [LPS99] Regina Y. Liu, Jesse M. Parelis, and Kesar Singh. Multivariate analysis by data depth: descriptive statistics, graphics and inference, (with discussion and a rejoinder by Liu and Singh). *Ann. Statist.*, 27(3):783–858, 06 1999.
- [Lyn07] Peter Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics* 227 (2008) 3431–3444, Február 2007.

- [Mic13] Microsoft. Microsoft visual basic for applications. <https://>, Január 2013. [Prístupné online: 19.1. 2013].
- [Mic15] Microsoft. Microsoft Excel. <https://products.office.com/en-us/excel>, 2015. [Prístupné online: 18.3.2015].
- [MP11] Andrew Vande Moere and Helen Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, Október 2011.
- [MS93] Michael C. Minnotte and David W. Scott. The mode tree: A tool for visualization of nonparametric density features. *Journal of Computational and Graphical Statistics*, 2:51–68, 1993.
- [NCEP14] NCEP. Inventory of Data Products on the NOAA Servers. <http://www.nco.ncep.noaa.gov/pmb/products/>, November 2014. [Prístupné online: 10.11.2014].
- [Nur03] Pertti Nurmi. *Recommendations on the verification of local weather forecasts*. European Centre for Medium Range Weather Forecasts, Decmeber 2003.
- [NWS15] NWS National Weather Service. The Ensemble Verification System (EVS). <http://amazon.nws.noaa.gov/ohd/evs/evs.html>, Marec 2015. [Prístupné online: 23.3. 2015].
- [OGJ11] P. Oldenburg, J.Halley Gotway, and T. Jensen. Model Evaluation Tools (MET) verification statistics visualization, 2011. *METViewer*.
- [Ope13] Apache OpenOffice. Openoffice.org basic programming guide. https://wiki.openoffice.org/wiki/Documentation/BASIC_Guide, Január 2013. [Prístupné online: 19.1. 2013].
- [Ope15] Apache OpenOffice. OpenOffice.org Calculate. <https://www.openoffice.org/product/calc.html>, 2015. [Prístupné online: 18.3.2015].

- [Par62] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):pp. 1065–1076, 1962.
- [Pea95] K. Pearson. Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Royal Society of London Philosophical Transactions Series A*, 186:343–414, 1895.
- [Poc11] Matthew Pocernich. *Forecast Verification: A Practitioner’s Guide in Atmospheric Science*, chapter Appendix - Verification Software, pages 232–240. John Wiley & Sons, Ltd., 2nd edition, December 2011.
- [Pot06] Kristin Potter. Methods for presenting statistical information: The box plot. In Hans Hagen, Andreas Kerren, and Peter Dannenmann, editors, *Visualization of Large and Unstructured Data Sets*, volume S-4 of *GI-Edition Lecture Notes in Informatics (LNI)*, pages 97–106. 2006.
- [Ros56] Murray Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832–837, September 1956.
- [Sco92] David W. Scott. *Multivariate density estimation : theory, practice, and visualization*. Wiley series in probability and mathematical statistics : Applied probability and statistics section. Wiley-Interscience, New York, Chichester, Brisbane, 1992.
- [SG11] Ying Sun and Marc G. Genton. Functional Boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334, Jún 2011.
- [SMY⁺05] Takafumi Saito, Hiroko Nakamura Miyamura, Mitsuyoshi Yamamoto, Hiroki Saito, Yuka Hoshiya, and Takumi Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS ’05, pages 23–, Washington, DC, USA, 2005. IEEE Computer Society.

- [SOA04] Neil C. Schwertman, Margaret Ann Owens, and Robiah Adnan. A simple more general boxplot method for identifying outliers. *Computational Statistics & Data Analysis*, 47(1):165–174, 2004.
- [Sol15] Exelis Visual Information Solutions. Interactive Data Language. <http://www.exelisvis.com/ProductsServices/IDL.aspx>, Marec 2015. [Prístupné online: 19.3. 2015].
- [SS07] Hideaki Shimazaki and Shigeru Shinomoto. A method for selecting the bin size of a time histogram. *Neural Comput.*, 19(6):1503–1527, June 2007.
- [Tay01] Karl E. Taylor. Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research*, 106(D7):7183–7192, April 2001.
- [Tay05] Karl E. Taylor. Taylor Diagram Primer. January 2005.
- [TM15] Inc. The MathWorks. Matlab the language of technical computing. <http://www.mathworks.com/products/matlab/>, Marec 2015. [Prístupné online: 18.3. 2015].
- [Tuk77] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [Tur93] Berwin A. Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*, 1993.
- [UCA15a] UCAR. NCAR Command Language. <http://www.ncl.ucar.edu/>, Marec 2015. [Prístupné online: 19.3. 2015].
- [UCA15b] UCAR/Unidata. NetCDF. <http://www.unidata.ucar.edu/downloads/netcdf/>, 2015. [Prístupné online: 18.3.2015].
- [Vas98] Tim Vasquez. *Observer Handbook*. International Weather Watchers, 1995, 1998.

- [VKMK11] Bahtijar Vogel, Arianit Kurti, Marcelo Milrad, and Andreas Kerren. An interactive web-based visualization tool in action: User testing and usability aspects. In *CIT*, pages 403–408. IEEE Computer Society, 2011.
- [VWvH⁺07] Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. Manyeyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, November 2007.
- [WBWK00] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pages 110–119, New York, NY, USA, 2000. ACM.
- [Wei14] Eric W. Weisstein. Statistical Median.
<http://mathworld.wolfram.com/StatisticalMedian.html>, 2014.
- [WMO94] WMO. *A GUIDE TO THE CODE FORM FM 92-IX Ext. GRIB Edition 1*. WMO, Máj 1994.
- [WMO03] WMO. *Introduction to GRIB Edition 1 and GRIB Edition 2*. WMO, Jún 2003.
- [WS12] Hadley Wickham and Lisa Stryjewski. 40 years of boxplots. Technical report, had.co.nz, 2012.