UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

PREDIKCIA ČASOVÝCH RADOV POMOCOU METÓD STROJOVÉHO UČENIA V OBLASTI POČÍTAČOVEJ BEZPEČNOSTI

Bakalárska práca

2023 **Alex Gajdoš**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

PREDIKCIA ČASOVÝCH RADOV POMOCOU METÓD STROJOVÉHO UČENIA V OBLASTI POČÍTAČOVEJ BEZPEČNOSTI

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	9.2.9. Aplikovaná informatika
Školiace pracovisko:	Ústav informatiky
Vedúci práce:	RNDr. Richard Staňa
Konzultant:	doc. RNDr. JUDr. Pavol Sokol, PhD.

Košice 2023 **Alex Gajdoš**

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Alex Gajdoš
Študijný program: aplikovaná informatika (jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: Informatika
Typ záverečnej práce: Bakalárska práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Predikcia časových radov pomocou metód strojového učenia v oblasti počítačovej bezpečnosti

Názov EN: Time series forecasting using machine learning in field of computer security

Cieľ:
1. Analyzovať vybrané existujúce prístupy predikcie časových radov.
2. Implementovať modely strojového učenia na predikciu časových radov v oblasti počítačovej bezpečnosti.
3. Porovnať dosiahnuté výsledky s existujúcimi výsledkami.

Literatúra:
1) Stana, R., Patrik, P., Gajdos, A., Pavol, S.: Network security situation awareness forecasting
based on neural networks. 8th International conference on Time Series and Forecasting (Submitted)
2) Pavol, S., Stana, R., Gajdos, A., Patrik, P.: Network security awareness forecasting based on statistical approach and neural networks. Logic Journal of IGPL (Submitted)

Vedúci: RNDr. Richard Staňa
Konzultant: doc. RNDr. JUDr. Pavol Sokol, PhD.
Oponent: Mgr. Viktor Pristaš
Ústav : ÚINF - Ústav informatiky
Riaditeľ ústavu: doc. RNDr. Ondrej Krídlo, PhD.

Dátum schválenia:

Poďakovanie

Rád by som touto cestou vyjadril úprimné a srdečné poďakovanie vedúcemu mojej bakalárskej práce RNDr. Richardovi Staňovi. Bol pre mňa veľkou oporou a pomohol mi svojimi vedomosťami a skúsenosťami pri spracovaní práce. Jeho odborné rady a konštruktívna kritika mi veľmi pomohli pri zdokonaľovaní mojich schopností a vedomostí pri práci. Taktiež by som sa chcel poďakovať mojej rodine, ktorá mi bola vždy oporou a podporou počas celého štúdia. Vďaka ich podpore som mohol venovať čas a energiu na vypracovanie tejto práce a splnenie si svojho cieľa.

Abstrakt

Táto bakalárska práca sa zameriava na predikciu časových radov v oblasti informačnej bezpečnosti, s cieľom poskytnúť sieťovým administrátorom lepší prehľad o aktuálnom stave siete a pomôcť im predvídať, ako sa tento stav bude meniť v budúcnosti. Sústreďuje sa na použitie metód strojového učenia, ako sú SARIMA, LightGBM, XGBoost, SVR a Prophet a porovnáva výsledky týchto metód s výsledkami prác, ktoré použili iné štatistické metódy a neurónové siete. Úloha predikcie je v tejto oblasti veľmi náročná z viacerých dôvodov, ako je nedostatok kvalitných dát a ich zložitosť v interpretácii. Cieľom práce je nájsť dobrý prístup v predikcii časových radov v oblasti informačnej bezpečnosti a tým pomôcť komunite v ďalšom výskume v tejto oblasti. Výsledky by mohli pomôcť sieťovým administrátorom a výskumníkom zlepšiť predikciu a poskytnúť lepší prehľad o aktuálnom stave a budúcich trendoch v oblasti sieťového bezpečnostného situačného povedomia.

Kľúčové slová: *sieťové bezpečnostné situačné povedomie, predikcia časových radov, strojové učenie.*

Abstract

This Bachelor's thesis focuses on the prediction of time series in the field of information security with the aim of providing network administrators with a better overview of the current state of the network and helping them anticipate how this state will change in the future. It concentrates on the use of machine learning methods, such as SARIMA, LightGBM, XGBoost, SVR and Prophet and compares their results with those obtained by studying how statistical methods and the neural networks affect the forecasting. The task of prediction in this field is very challenging for several reasons, including a lack of quality data and their complexity in interpretation. The aim of this work is to find a good approach in the prediction of time series in the field of information security and thereby help the community in further research in this area. The results could help network administrators and researchers improve prediction and provide a better overview of current state and future trends in the field of network security situational awarenes.

Keywords: *network security situational awareness, time series forecasting, machine learining.*

Obsah

Úvod

V dnešnej dobe, kedy sa kybernetické hrozby stávajú stále sofistikovanejšími, ochrana informačných systémov pred nimi je stále zložitejšia a náročnejšia. Sieťoví administrátori sa musia neustále prispôsobovať novým spôsobom a technikám útokov, aby zabezpečili bezpečnosť informačných systémov. Predikcia časových radov pomocou metód strojového učenia sa javí ako jedna z účinných metód, ktorá môže pomôcť v boji proti týmto hrozbám. Táto metóda umožňuje vytvoriť modely na základe historických údajov a predpovedať budúce hodnoty. Tieto predpovede môžu slúžiť ako varovania pred nezvyčajnými udalosťami a pomáhať pri identifikácii potenciálnych útokov.

V tejto práci sa zameriavame na predikciu časových radov pomocou metód strojového učenia. V súčasnosti, kedy počítačové hrozby prezentujú závažné riziko pre informačné systémy, má predikcia časových radov v oblasti kybernetickej bezpečnosti kľúčový význam. Je dôležite vedieť predpokladať, ako sa bude vyvíjať situácia v sieti a na základe toho prijať preventívne opatrenia a chrániť systémy pred útokmi. Metódy strojového učenia sa stávajú čoraz dôležitejšími v oblasti kybernetickej bezpečnosti, keďže umožňujú predikovať budúce hodnoty založené na histórii. Tieto metódy zahŕňajú napríklad regresnú analýzu alebo analýzu dekompozície časových radov. Avšak, predikcia časových radov v oblasti kybernetickej bezpečnosti predstavuje výzvu, pretože počítačové hrozby sa neustále menia a prispôsobujú sa novým technológiám. Preto je kľúčové využiť najnovšie algoritmy strojového učenia, ktoré umožňujú identifikovať aj najjemnejšie zmeny v časových radoch a poskytujú presnejšie predpovede. V konečnom dôsledku, predikcia časových radov je nástroj, ktorý môže byť veľmi užitočný pri zabezpečovaní informačných systémov pred počítačovými hrozbami. V rámci tejto práce sa zameriavame na konkrétny aspekt kybernetickej bezpečnosti, a to na sieťové bezpečnostné situačné povedomie. Ide o kritickú zložku v oblasti počítačovej bezpečnosti, ktorá poskytuje správcom siete aktuálny pohľad na stav siete a pomáha identifikovať akékoľvek anomálie alebo hrozby. Cieľom je rýchlo identifikovať, analyzovať a reagovať na bezpečnostné incidenty, aby sa minimalizovali ich negatívne následky.

V prvom kroku táto práca analyzuje existujúce prístupy predikcie časových radov pomocou metód strojového učenia v oblasti počítačovej bezpečnosti. Zameriava sa

na výhody a nevýhody jednotlivých prístupov a určenie prístupu, ktorý by bol najvhodnejší pre tento konkrétny prípad. Analyzovať bude napríklad časové rady týkajúce sa útokov na sieťové zariadenia. Po analýze existujúcich prístupov bude nasledovať implementácia modelov strojového učenia na predikciu časových radov. Tieto modely budú trénované na základe existujúcich datasetov časových radov v oblasti kybernetickej bezpečnosti. V poslednom kroku bude táto práca porovnávať dosiahnuté výsledky s existujúcimi výsledkami získanými z iných prác v oblasti predikcie časových radov. Diskutovať bude o výhodách a nevýhodách použitých modelov a o tom, aké možnosti na zlepšenie existujú.

Práca sa skladá zo 4 kapitol. Prvé dve kapitoly sú zamerané na teoretické poznatky z predikcie časových radov v oblasti kybernetickej bezpečnosti. Tretia kapitola sa zameriava na vybrané metódy strojového učenia na predikciu časových radov v tejto oblasti. V poslednej kapitole prezentujeme dosiahnuté výsledky a porovnávame ich s existujúcimi prístupmi.

1 Predikcia v oblasti kybernetickej bezpečnosti

S narastajúcim počtom kybernetických útokov sa kybernetická bezpečnosť stáva čoraz dôležitejšou. Predikcia útokov a bezpečnostných incidentov je kľúčová pre prevenciu a ochranu proti týmto hrozbám. V prehľadovom článku [16] sa úloha predikcie delí na tri prípady. Podrobný prehľad tohto rozdelenia sú v Tabuľke č. 1. Prvým prípadom tohto rozdelenia je projekcia útoku, ktorá sa zameriava na identifikáciu nasledujúcich krokov počas útoku útočníka a odhalenie jeho finálneho cieľa útoku. Druhým prípadom tohto rozdelenia je predikcia útoku, ktorého hlavnou úlohou je predikovať útok ešte predtým než sa to stane. Tretím a zároveň aj posledným prípadom tohto rozdelenia je predikcia sieťovej bezpečnostnej situácie, ktorý je veľmi všeobecný a súvisí s kybernetickým situačným povedomím. Ide o komplexnú úlohu, ktorá sa zameriava na predpovedanie situácie v celej sieti. Cieľom tejto predikcie je zlepšenie reakcie na bezpečnostné incidenty a poskytnutie lepšieho prehľadu o stave siete a jej rizikách. Všetky tieto prípady si rozoberieme v nasledujúcej časti, kde si podrobnejšie popíšeme ich hlavnú úlohu, podobne ako v prehľadovom článku [16].

prípad	úloha	ďalší prehľad
projekcia útoku rozpoznanie úmyslu útočníka	identifikácia nasledujúcich krokov útočníka finálneho cieľa útočníka	[44, 2]
predikcia útoku	predpovedať kedy, kde a aký druh útoku nastane	[1]
predikcia sieťovej bezpečnostnej situácie	predikcia vývoju celkovej situácie v sieti	[25]

Tabuľka 1: Tabuľka oblastí predikcie v kybernetickej bezpečnosti. [16]

1.1 Projekcia útoku

Počiatok myšlienky projekcie útoku sa datuje do roku 2001, keď Geib a Goldman [14] navrhli rozšírenie rozpoznávania plánu útoku a nazvali ho projekciou útoku. Zároveň identifikovali predpoklady a možné problémy, ako napríklad nutnosť pracovať s nepozorovanými akciami, neúspech pri pozorovaní a zohľadnenie viacerých súčasných cieľov. Projekcia útoku je technika používaná v kybernetickej bezpečnosti na predpovedanie potenciálnych útokov a ich dopadov na systémy alebo organizácie. Táto technika umožňuje identifikovať zraniteľnosti v systéme, predpovedať, aké druhy útokov by mohli byť použité a aký by bol ich potenciálny vplyv. Existuje mnoho publikácií, ktoré sa zaoberajú projekciou útokov a rôznymi aspektami tejto techniky. Prvé metódy projekcie útoku sa začali objavovať okolo roku 2003 v prácach [15, 34], avšak výskum v tejto oblasti je stále aktívny. Jednou z prvých metód bolo použitie Dynamických Bayesovských sietí (DBN), ktoré využil Qin a Lee v roku 2004. Qin a Lee boli medzi prvými, ktorí navrhli schému projekcie útoku na vysokej úrovni. Ich systém bol navrhnutý na prispôsobenie senzorových pozorovaní preddefinovaným štruktúram útokov pomocou DBN. Tento prístup umožňuje definovanie príčinno-dôsledkových vzťahov medzi pozorovanými udalosťami a čo je ešte dôležitejšie, dynamické učenie sa pravdepodobností prechodu pomocou dostatočného množstva dát. Po naučení sa pravdepodobností prechodu, môžu byť tieto pravdepodobnosti použité na predpovedanie potenciálnych budúcich útokov, ako je uvedené v článku [44]. Vo všeobecnosti však platí, že na to, aby sme boli schopní identifikovať nasledujúce kroky útočníka a pokúsiť sa ich predikovať, potrebujeme najprv vedieť bežné správanie útočníkov. Príklad takejto postupnosti krokov útoku je v práci [48], kde útočník zvolil tento postup:

1. útočník identifikuje cieľový systém a získava informácie o jeho zraniteľnostiach a slabých miestach,
2. útočník vyberie vhodný druh útoku, ktorý by mohol byť použitý na získanie prístupu k cieľovému systému alebo na iné škodlivé účely,
3. útočník rozširuje škodlivý kód na cieľovom systéme a vykonáva útok,
4. škodlivý kód získava kontrolu nad cieľovým systémom a útočník má prístup k dôležitým dátam alebo môže spôsobiť inú formu škody,
5. útočník zahladí dôkazy útoku.

Zjednodušený postup krokov bol uvedený aj v tejto práci [16]:

1. kybernetické skenovanie,

2. enumerácia,

3. pokus o vniknutie,

4. eskalácia privilégií,

5. vykonávanie škodlivých operácií,

6. nasadenie škodlivého kódu/zadných vrátok,

7. likvidácia forenzných dôkazov a opustenie systému.

Takýto postup je možné sledovať na základe známych krokov útoku. Existuje mnoho rôznych typov útokov, a preto je potrebné vytvoriť model, ktorý bude použiteľný na predpovedanie viacerých útokov. Historicky prvými modelmi boli tie, ktoré boli vytvorené manuálne, ale moderné metódy sa sústreďujú na automatické generovanie modelov pomocou dolovania dát. Rozpoznávanie zámeru útočníka je podobný koncept ako projekcia útoku, ale sústreďuje sa na motiváciu útočníka. Ak odhalíme cieľ útočníka, môžeme predpokladať budúce bezpečnostné udalosti podľa konkrétneho útoku. Nové techniky sa snažia rozpoznávať zámer útočníka v reálnom čase a stále viac sa približujú k projekcii útoku. V literatúre sa tento koncept rozpoznávania zámeru útočníka často spomína ako rozpoznávanie hrozieb (threat recognition) alebo detekcia útokov (attack detection).

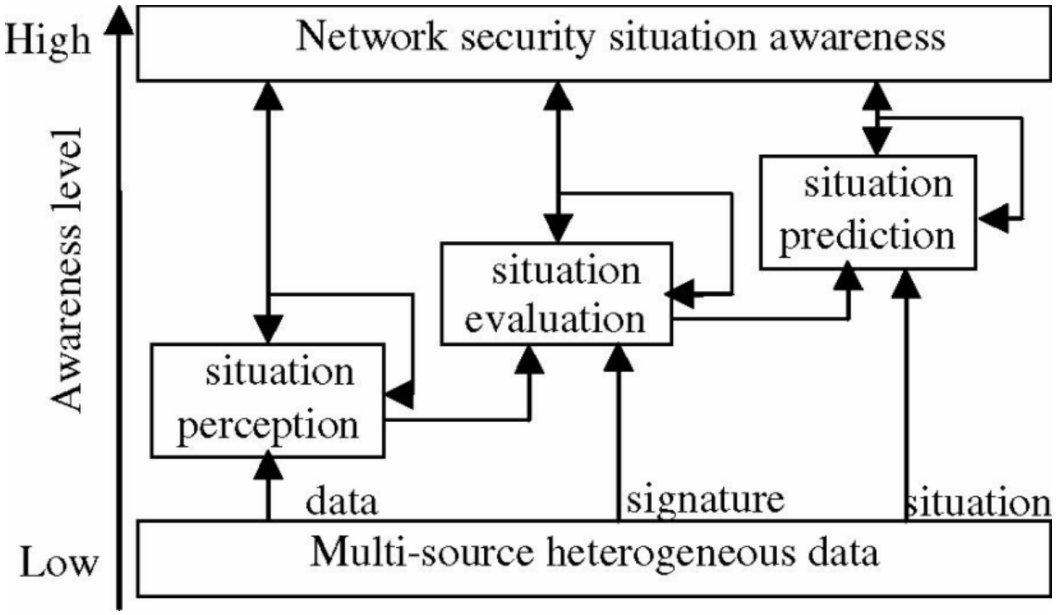
1.2 Predikcia útoku

Samotná predikcia útokov predstavuje významnú úlohu v oblasti kybernetickej bezpečnosti, najmä pri pokusoch o prienik do systému [26]. V porovnaní s projekciou útokov, ktorá sa snaží analyzovať a odhaliť existujúci útok, predikcia útokov sa sústredí na predpovedanie nového útoku. Vzhľadom na všeobecný charakter úlohy v súčasnosti neexistuje jednotný prístup k jej riešeniu [5]. Metódy projekcie útokov sa najčastejšie spoliehajú na diskrétne modely kybernetických útokov [24]. Na druhej strane, metódy a modely používané pre predikciu útokov sú rôznorodé: od diskrétnych, ako napr. graf útokov po spojité, ako napr. časové rady [5]. Diskrétny model použitý pre projekciu útokov môže byť zmenený a použitý aj pre predikciu, pričom predikcia útoku sa v tomto prípade nezačína s aktuálne prebiehajúcou škodlivou udalosťou, ale s pravdepodobnosťou, že sa objaví konkrétna zraniteľnosť v počítačovej sieti [46]. Spojitý model založený na časových radoch počtu útokov na konkrétny systém alebo sieť v čase môže byť použitý na predikciu toho, či sa útok uskutoční alebo nie [9]. Pokročilejšie metódy môžu pracovať s konkrétnymi typmi útokov, charakteristikou útočníka i obete, a tak

môžu odhadnúť, aký typ útoku nastane, kto bude útočníkom a kto bude cieľom útoku [5]. Novšie metódy často využívajú aj iné zdroje dát na predikciu útokov, ako napríklad informácie zo sociálnych sietí alebo sledujú zmeny v správaní používateľov [23].

1.3 Predikcia sieťovej bezpečnostnej situácie

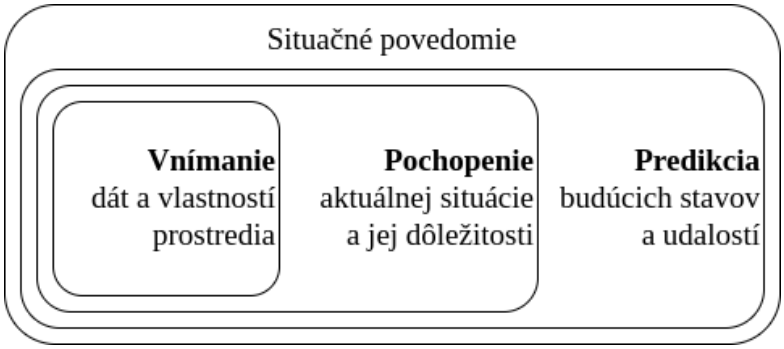
Predikcia sieťovej bezpečnostnej situácie sa zameriava na aktuálny globálny stav systému alebo siete. Tento prístup sa často označuje ako kybernetické situačné povedomie (CSA) alebo ako sieťové bezpečnostné situačné povedomie (NSSA). NSSA vychádza z konceptu situačného povedomia, ktorý vznikol vo vojenskom výskume. Samotne NSSA si bližšie popíšeme na Obrázku 1, ktorý je v práci [21] bližšie vysvetlený.



Obr. 1: Koncepčný model NSSA Zdroj [21].

Najširšie používaná definícia situačného povedomia je z práce [12]: Vnímanie prvkov v prostredí v čase a priestore, pochopenie ich významu a predikcia ich hodnoty v blízkej budúcnosti. Definícia popisuje tri stupne situačného povedomia: vnímanie, pochopenie a predikcia, ako je zobrazené na Obrázku 2. Predikcia sa zameriava na predpoveď zmien v kybernetickej bezpečnostnej situácii a jej významnosť je hlboko zakorenená v teoretickom pozadí situačného povedomia. Väčšina prác používa kvantitatívnu analýzu na popis sieťovej bezpečnostnej situácie v určitom čase, kde výsledné hodnoty sa následne použijú na predpovedanie budúcej situácie. Takýto prístup nedáva žiadnu pridanú informáciu o presnej povahe budúcich útokov, no môže upozorniť na zlepšenie alebo zhoršenie celkovej bezpečnostnej situácie v sieti. Kvantitatívny prístup

umožňuje použitie metód na analýzu a predikciu, ktoré sú preskúmané a používané v iných oblastiach. Kvantitatívna analýza vyžaduje metriku na vyhodnotenie sieťovej bezpečnostnej situácie, no zatiaľ neexistuje žiadna zavedená a štandardne používaná. Bližšie podrobnosti sú uvedené v práci [20].



Obr. 2: Stupne situačného povedomia. Zdroj [13].

Ak sa na to pozrieme z pohľadu kybernetickej bezpečnosti, tak v publikácii [19] sú stupne situačného povedomia vysvetlené takto: vnímanie predstavuje znalosť prvkov v sieti, ako sú výstrahy hlásené detekčným systémom, protokoly firewallu, správy o skenovaní, ako aj čas, kedy sa udiali. Pochopenie sa definuje ako súbor techník, metodológií, procesov a postupov, ktoré bezpečnostní analytici používajú na analýzu, syntézu, koreláciu a agregáciu dôkazových údajov vnímaných v sieti z rôznych prvkov siete. Bezpečnostná vizualizácia, ktorá je súčasťou fázy pochopenia, je prevedenie organizovaných dát a informácií do zmysluplných vzorov alebo postupov na to, aby boli vizualizované. Fúzia dát je technika na agregáciu sád dôkazov týkajúcich sa prebiehajúcej situácie. Predikcia je schopnosť predpovedať budúce udalosti na základe poznatkov získaných z dynamiky prvkov siete a pochopenia situácie.

2 Predikcia časových radov

Časový rad je súbor hodnôt, ktoré sú chronologicky usporiadané v čase a umožňujú tak zaznamenať takmer akýkoľvek merateľný údaj. Medzi tieto údaje patria veličiny ako teplota, vlhkosť, rýchlosť vetra a podobne, s ktorými sa stretávame v každodennom živote, napríklad pri predpovedi počasia.

V súčasnej dobe majú tieto rady veľmi aktuálne využitie v oblasti ekonomiky, napríklad pri pravidelne zvyšujúcej sa inflácii, ktorej vývoj môžeme práve takýmto spôsobom sledovať. V ekonomike sa využitie časových radov stáva čoraz dôležitejším, vďaka čomu existuje množstvo metód a nástrojov na analýzu časových radov a ich použitie v predikcii budúcich trendov [41]. Časové rady majú veľmi široké využitie v rôznych oblastiach, nielen v ekonomike. Napríklad v medicíne sa často používajú na sledovanie vývoja chorôb a stavu pacientov v čase. V sociológii sa často používajú na sledovanie trendov v spoločnosti, ako napríklad počet manželstiev alebo rozvodov.

Pre lepšie pochopenie vzorov v dátach a pre vytváranie predpovedí o budúcich hodnotách je dôležitá identifikácia zložiek, ako sú trend, sezónnosť, cykly a rezíduá. Podľa [38] je táto identifikácia kľúčová pre úspešné využitie časových radov v predpovedaní budúcich trendov a zlepšenie prognóz. Kvalita predikcie závisí od mnohých faktorov a zohľadňuje celý proces od výberu vhodného modelu a parametrov až po interpretáciu výsledkov. Jedným z kľúčových faktorov je kvalita a množstvo vstupných dát, na základe ktorých sa predikcia robí. Ďalšie faktory zahŕňajú:

- Výber vhodného modelu: Existuje množstvo modelov, ktoré sa dajú použiť na predikciu, ako napríklad ARIMA, VAR, či LSTM. Výber správneho modelu je kritickým faktorom v predikcii, pretože rôzne modely môžu byť vhodné na predikciu rôznych typov časových radov.

- Kvalita trénovacej sady: Trénovacia sada je množina dát, ktoré sa používajú na učenie modelu. Je dôležité zvoliť dostatočne veľkú trénovaciu sadu, aby model mal dostatok dát na učenie a zvládol zaznamenať zložité vzorce v dátach a taktiež si je potrebné dať pozor, aby sme model nepreučili.

- Vyhodnocovanie modelu: Vyhodnocovanie modelu je kritickou súčasťou predik-

cie. Existuje mnoho metrík, ktoré sa používajú na hodnotenie kvality predikcie, ako napríklad MAE, RMSE, či R2. Výber správnej metriky závisí od charakteru predikovaných dát a cieľov predikcie.

- Výber parametrov: Pri každom modeli sa používajú určité parametre, ktoré treba zvoliť. Tieto parametre ovplyvňujú výkon modelu a správny výber parametrov môže viesť k lepším výsledkom.

Predikcia sa zvyčajne skladá z nasledujúcich krokov:

1. Zhromažďovanie dát: Potrebné sú dáta vo forme časových radov, ktorých zdroj je široko dostupný napríklad vo verejných databázach, ekonomických štatistikách a podobne.

2. Spracovanie dát: Dáta musia byť upravené a očistené od chýb a nekonzistentností. Tiež sa často používajú techniky ako normalizácia a škálovanie, aby sa zabezpečila kompatibilita medzi dátami.

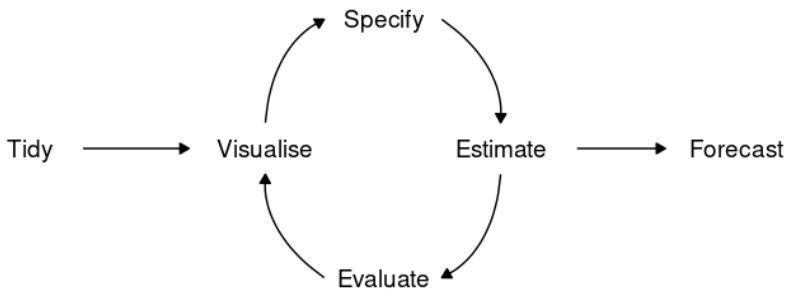
3. Výber modelu: Pri výbere modelu treba zohľadniť charakteristiku časového radu a zvoliť model, ktorý je najvhodnejší na jeho predpokladanú dynamiku. Existuje veľa modelov, ako napríklad ARIMA, autoregresia s exogénnymi premennými (ARX), autoregresia s kĺzavými priemermi (ARMA) a mnoho ďalších.

4. Odhad parametrov: Pri použití konkrétneho modelu je potrebné odhadnúť jeho parametre, čo obvykle zahŕňa priradenie konkrétnych hodnôt parametrov na základe trénovacej sady dát.

5. Overenie modelu: Model sa musí overiť na základe testovacieho súboru dát, aby sa zistila jeho presnosť a schopnosť predikcie.

6. Predikcia: Nakoniec sa model použije na predpovedanie budúcich hodnôt a vytvorenie časového radu predikovaných hodnôt.

Tento proces je zobrazený na Obrázku č. 3. Podľa [17] je predikcia procesom odhadovania budúcich hodnôt na základe minulých hodnôt a môže byť použitá v mnohých oblastiach, ako napríklad v ekonomike, medicíne, či meteorológii. Dôležité je si uvedomiť, že predikcie časového radu nie sú vždy presné a môžu byť ovplyvnené rôznymi faktormi, ako napríklad šumom v dátach. V nasledujúcej časti bližšie popíšeme trend, sezónnosť, cykly a rezíduá vzhľadom na to, že sú dôležitou súčasťou predikcie časových radov.



Obr. 3: Popis procesu predikcie od tvorby dát až po samotnú predikciu. Zdroj [18].

2.1 Základné komponenty časových radov a ich analýza

Trend v časových radoch predstavuje dlhodobý pohyb dát v čase a je jeden z kľúčových faktorov, ktoré ovplyvňujú časové rady a jeho identifikácia a modelovanie je dôležité pre presnú predpoveď budúcich hodnôt. Podľa [17] je trend dlhodobý pohyb v časových radoch, ktorý predstavuje podkladový vzor alebo smer dát v čase. Trend môže byť rastúci, klesajúci alebo sa môže pohybovať vodorovne. Identifikácia trendu v časových radoch môže byť problematická, pretože v reálnych dátach sa často vyskytujú fluktuácie a náhodný šum, ktoré môžu skresliť výsledky. Existuje však viacero metód, ktoré sa používajú na identifikáciu trendu, ako napríklad lineárna regresia, exponenciálne vyhladzovanie a rôzne časové modely. Tieto metódy sa používajú na prispôsobenie čiary alebo krivky dátam, ktoré sa potom dajú použiť na robenie predpovedí budúcich hodnôt. Niekedy môže byť zložité rozlíšiť trend od sezónnosti, pretože sezónne vplyvy môžu byť veľmi silné a často sa prekrývajú s trendom. Preto sa často používajú metódy rozkladu časových radov na jeho komponenty, ktoré umožňujú identifikovať trend, sezónnosť a šum [8]. Trend sa v časových radoch často považuje za dôležitý ukazovateľ pre mnohé odvetvia, ako napríklad ekonomiku. V tomto kontexte sa často používa na predpovedanie budúcich vývojov na trhu a v investičných príležitostiach. Taktiež sa používa v predpovednej analýze v rôznych oblastiach, ako napríklad v marketingu, predajoch a výrobnom plánovaní.

Sezónnosť v časových radoch sa týka opakujúcich sa vzorov v dátach, ktoré sa vyskytujú v pravidelných časových obdobiach. Tieto vzory môžu byť spôsobené rôznymi faktormi, ako sú sezónne zmeny, sviatky alebo iné udalosti. Tento vzor môže byť zrejmý pri vizuálnom pohľade na časový rad, kde vidíme, že hodnoty sa v určitom období v čase pravidelne zvyšujú alebo znižujú. Sezónnosť sa často modeluje pomocou

tzv. sezónneho indexu, ktorý reprezentuje pomerné zvýšenie alebo zníženie hodnôt v danom období v porovnaní s ostatnými obdobiami. Sezónny index sa dá vypočítať pomocou rôznych metód, napríklad pomocou spriemerovania hodnôt v rovnakom období za predchádzajúce roky [17]. V časových radoch môže byť sezónnosť kombinovaná s inými komponentami, ako je trend alebo cyklickosť. Jednou z metód rozkladu časových radov je tzv. aditívny model, ktorý sa skladá z komponentov trendu, sezónnosti a šumu. Tento model sa dá vyjadriť nasledujúcou rovnicou:

$$y(t) = T(t) + S(t) + e(t)$$

kde y(t) je hodnota v čase t, T(t) je trend, S(t) je sezónny index a e(t) je šumová zložka [6]. Je dôležité si uvedomiť, že sezónnosť nie je vždy prítomná v časových radoch a aj keď je prítomná, môže sa líšiť v intenzite a dĺžke. Preto je potrebné ju dôkladne analyzovať a modelovať, aby sme mohli robiť správne predpovede budúcich hodnôt.

Cyklickosť v časových radoch sa týka opakujúcich sa, no nie pravidelných vzorov v dátach, ktoré majú vplyv na dlhodobé trendy. Vyskytuje sa vo väčšine hospodárskych cyklov a môže byť spôsobená mnohými faktormi, ako napríklad výkyvmi v ekonomickej činnosti, zmenami módnych trendov, prírodnými katastrofami a podobne. V publikácii [17] bolo spomenuté, že identifikácia cyklickosti v časových radoch môže byť zložitá, pretože cykly môžu byť nepravidelné a môžu sa líšiť v dĺžke a intenzite. Jedným zo spôsobov, ako identifikovať cyklickosť v dátach, je použitie ekonometrických modelov, ako napríklad Hodrick-Prescottovho filteru alebo Kalmanovho filtra. Tieto modely dokážu oddeliť cyklické komponenty od trendu a sezónnosti a umožňujú tak lepšie pochopenie pohybu dát v čase. Podobne ako sezónnosť, aj cyklickosť môže mať významný vplyv na predikciu budúcich hodnôt v časových radoch. Preto je dôležité, aby boli cyklické vzory zahrnuté do modelovania a predikcie budúcich trendov.

Šum v časových radoch sa vzťahuje na náhodné variácie alebo kolísanie v dátach, ktoré sa vymykajú vzoru alebo trendu. Tieto variácie môžu byť spôsobené rôznymi faktormi, ako napríklad chyby merania, náhodné udalosti alebo odchýlky [8]. Jednou z kľúčových výziev pri zaobchádzaní so šumom v časových radoch je rozlíšenie ho od zmysluplných vzorov alebo trendov v dátach. To môže byť vykonané rôznymi technikami, ako napríklad dekompozícia časových radov, ktorá rozdeľuje dáta na jeho súčasti, vrátane trendu, sezónnosti a šumu [6]. Je dôležité rozlíšiť šum od skutočných vzorov a trendov v časových radoch, aby sme mohli presne predpovedať budúce hodnoty. Existuje niekoľko techník na odstránenie šumu z dát, z ktorých niektoré zahŕňajú filtračné techniky. Medzi tieto techniky patrí napríklad filtrovanie nízkych frekvencií, ktoré odstraňuje vysokofrekvenčný šum a zachováva nízkofrekvenčné vzory, a filtrovanie vysokých frekvencií, ktoré robí presný opak - odstraňuje nízkofrekvenčné vzory a zachováva vysokofrekvenčný šum [38].

3 Metódy strojového učenia

Strojové učenie možno všeobecne definovať ako výpočtové metódy, ktoré využívajú skúsenosti na zlepšenie výkonu alebo na vytváranie presných predpovedí [31]. Strojové učenie, resp. jeho algoritmy, sú v istom zmysle mäkko programované (soft coded) a to tak, že sa automaticky menia alebo prispôsobujú svoju architektúru opakovaním, t.j. skúsenosťami, aby sa stali lepšími v plnení požadovanej úlohy. Proces adaptácie sa nazýva trénovanie, pri ktorom sa poskytnú vzorky vstupných dát spolu s požadovanými výstupmi. V optimálnom prípade chceme algoritmus strojového učenia dostať do stavu, aby nielen produkoval požadovaný výstup, keď sú mu predstavené trénovacie vstupy, ale aby mohol generalizovať a produkovať požadovaný výstup aj z nových, predtým nevidených dát. Trénovanie nemusí byť obmedzené na prvotnú adaptáciu počas konečného intervalu. Podobne ako u ľudí, dobrý algoritmus môže praktizovať celoživotné učenie, keď spracováva nové dáta a učí sa z vlastných chýb [11]. Tak ako v iných oblastiach informatiky, aj tu sú čas a priestorová zložitosť kritické ukazovatele kvality týchto algoritmov. V strojovom učení však budeme navyše potrebovať vedieť zložitosť dát, aby sme mohli vyhodnotiť počet vzoriek, ktoré sú potrebné pre algoritmus na naučenie sa. Všeobecnejšie, záruky teoretického učenia pre algoritmus závisia od zložitosti uvažovaných tried a veľkosti trénovacích dát. Keďže úspešnosť algoritmu učenia závisí od použitých údajov, strojové učenie neodmysliteľne súvisí s analýzou údajov a štatistikou. Vo všeobecnosti techniky učenia sú metódy založené na údajoch, ktoré kombinujú základné pojmy v informatike s nápadmi zo štatistiky, pravdepodobnosti a optimalizácie [31]. S úspešným použitím strojového učenia sa spájajú aj dve najväčšie výhody. Prvou výhodou je, že môže nahradiť namáhavú a opakujúcu sa prácu človeka. Druhou výhodou a ešte významnejšou je, že sa môže potenciálne naučiť komplikovanejšie a jemnejšie vzorce vstupných dát, ako by bol schopný priemerne vnímavý ľudský pozorovateľ. Obe výhody sú napr. použité v rádioterapii. Algoritmus pre kontúrovanie môže zachytiť sofistikovanejšie a jemnejšie vzory vstupných dát než priemerný pozorovateľ, a to buď na jednom obrázku, alebo simultánne integrovať údaje z viacerých zdrojov. Tieto výhody sú pre radioterapiu veľmi dôležité, pretože napríklad denne kontúrovanie nádorov a orgánov počas plánovania liečby je

časovo náročný proces rozpoznávania vzorov, ktorý závisí na znalosti a skúsenostiach pozorovateľa s výskytom anatómie na diagnostických snímkach. Avšak táto znalosť má svoje limity a výsledné kontúry sú spojené s neistotou a variabilitou pozorovateľov [11]. Podľa publikácie [31] je strojové učenie schopné úspešne riešiť širokú škálu úloh, ktoré si teraz podrobnejšie opíšeme.

- Klasifikácia (Classification): ide o problém priradenia kategórie ku každej položke. Príklad klasifikácie pri dokumentoch môže pozostávať z priradenia kategórie, ako je politika, obchod, šport alebo počasie, ku každému dokumentu. Počet kategórií v takýchto úlohách je často menší ako niekoľko stoviek, ale v niektorých náročných úlohách môže byť oveľa väčší a dokonca neobmedzený, ako napríklad pri OCR, klasifikácii textu alebo rozpoznávaní reči.

- Hodnotenie (Ranking): ide o problém zoradenia položiek podľa nejakého kritéria. Vyhľadávanie na webe, napr. vrátenie webových stránok relevantných pre vyhľadávací dopyt, je to príklad kanonického hodnotenia. Mnoho ďalších podobných problémov s klasifikáciou vzniká v kontexte návrhu systémov extrakcie informácií alebo spracovania prirodzeného jazyka.

- Klastrovanie (Clustering): ide o problém rozdelenia množiny položiek do homogénnych podmnožín. Klastrovanie sa často používa na analýzu veľmi veľkých súborov údajov. Napríklad v kontexte analýzy sociálnych sietí sa klastrovacie algoritmy pokúšajú identifikovať prirodzené spoločenstvá v rámci veľkých skupín ľudí.

- Zníženie dimenzionality alebo mnohostranné učenie (Dimensionality reduction or manifold learning): problém spočíva v transformácii počiatočnej reprezentácie položiek na reprezentáciu nižšej dimenzie pri zachovaní niektorých vlastností počiatočnej reprezentácie. Bežným príkladom je predspracovanie digitálnych obrazov v úlohách počítačového videnia.

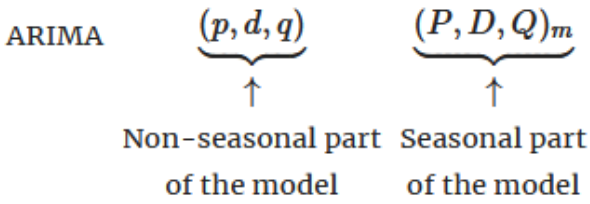
- Regresia (Regression): ide o problém predpovedania skutočnej hodnoty pre každú položku. Príklady regresie zahŕňajú predikciu hodnôt akcií alebo variácií ekonomických premenných. Pri regresii závisí penalizácia za nesprávnu predpoveď od veľkosti rozdielu medzi skutočnými a predpokladanými hodnotami, na rozdiel od problému klasifikácie, kde zvyčajne neexistuje žiadna predstava o blízkosti medzi rôznymi kategóriami.

V rámci našej práce sa zaoberáme rôznymi metódami strojového učenia, ktoré majú za cieľ predikovať hodnoty regresného problému. Medzi tieto metódy patrí napríklad SARIMA, LightGBM, Prophet, SVR a XGBoost. V nasledujúcich podkapitolách

sa budeme podrobne venovať každej z týchto metód a popíšeme ich princíp fungovania a vlastnosti.

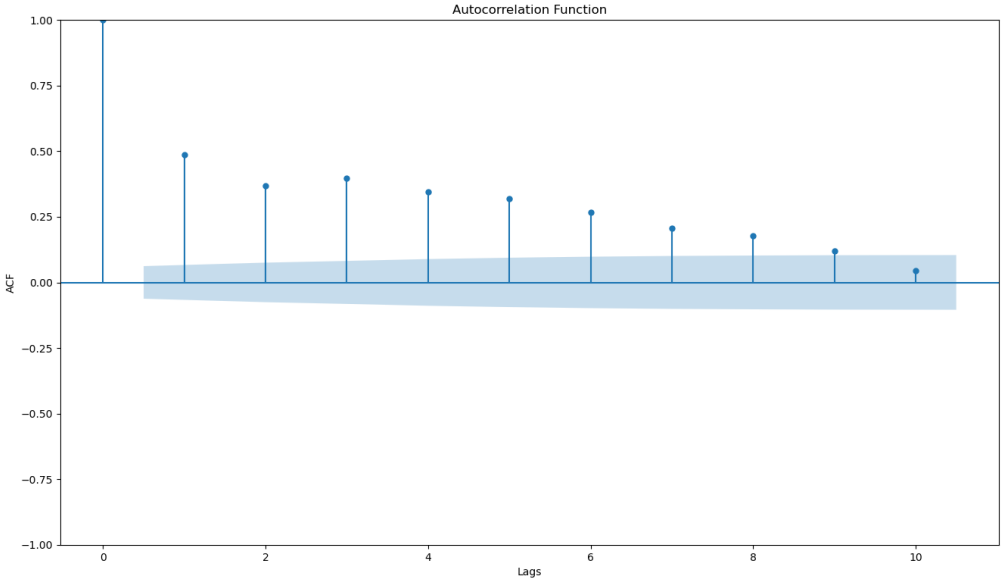
3.1 Seasonal ARIMA

ARIMA, teda Autoregressive Integrated Moving Average, predstavuje jednu z najbežnejšie využívaných metód na predikciu jednorozmerných časových radov. Aj keď táto metóda dokáže spracovať dáta s trendom, nie je schopná pracovať s časovými radmi obsahujúcimi sezónne zložky [7]. Preto existuje jej rozšírenie, ktoré umožňuje priame modelovanie sezónnej zložky v časových radoch, ktoré sa nazýva SARIMA (Seasonal Autoregressive Integrated Moving Average). Základná štruktúra modelu vyzerá takto:

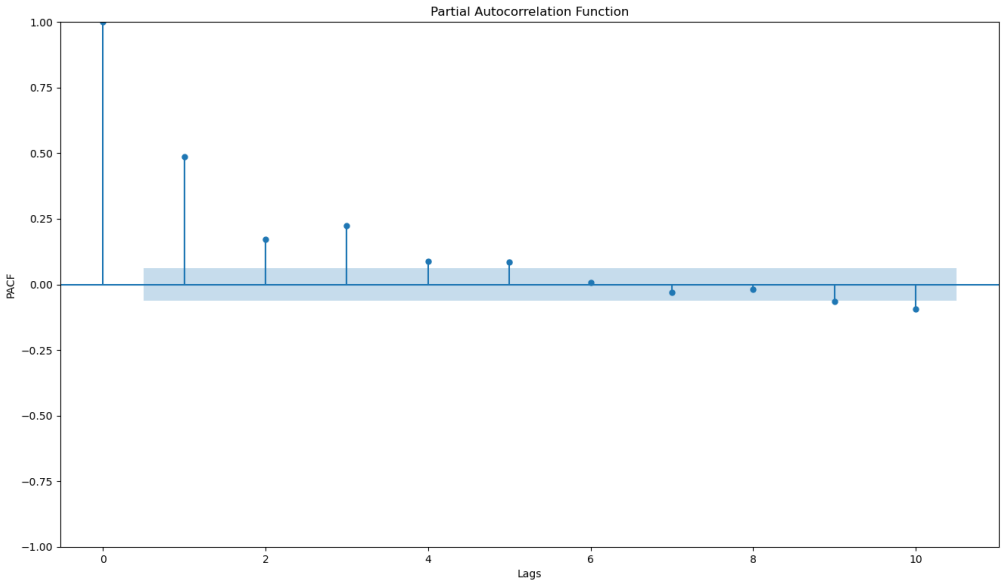


Obr. 4: Zdroj. [17]

kde p a P predstavujú nesezónne a sezónne stupne autoregresivnej časti (AR), q a Q predstavujú nesezónne a sezónne stupne kĺzavého priemeru (AR) a d a D predstavujú nesezónne a sezónne stupne diferencie (I) [27]. Posledný parameter sa v každej literatúre objavuje rozlične, najčastejšie sa však stretneme s označením m alebo s a tento parameter reprezentuje počet časových krokov za jedno sezónne obdobie. Pre správny výber parametrov sa používajú rôzne techniky no jedna z najčastejších je využite ACF a PAFC grafov. Autokorelačná funkcia (ACF) je korelácia medzi aktuálnou a minulými hodnotami rovnakej premennej. Tento prístup zohľadňuje aj časový efekt, ktorý sa prenáša na hodnoty a ovplyvňuje ich v čase, okrem priameho efektu. Napríklad ceny ropy 2 dni dozadu ovplyvnia ceny 1 deň dozadu a nakoniec aj dnes. Ale ceny ropy 2 dni dozadu by mohli mať tiež vplyv na dnes, čo ACF meria. Parciálna autokorelácia (PACF) na druhej strane meria iba priamu koreláciu medzi minulými hodnotami a aktuálnymi hodnotami. Napríklad PACF bude merať iba vplyv cien ropy 2 dni dozadu na dnes bez prenosového efektu.



Obr. 5: Ukažka ACF grafu pri lagu 12



Obr. 6: Ukažka PACF grafu pri lagu 12

V publikácii [17] sa uvádza, ze sezónna časť modelu AR alebo MA bude viditeľná v sezónnych oneskoreniach PACF a ACF grafu. Ak si predstavíme model s týmito

parametrami

$$ARIMA(0,0,0)(0,0,1)12$$

model sa ukáže nasledovne:

- hrot pri vrchole 12, ale žiadne ďalšie významne hroty pri iných vrcholoch
- exponenciálny pokles v sezónnych oneskoreniach PACF (pri vrcholoch 12, 24, 36, ...)

Podobne v modeli s parametrami

$$ARIMA(0,0,0)(1,0,0)12$$

uvidíme:

- exponenciálny pokles v sezónnych lagoch ACF
- jediný významný hrot pri vrchole 12 v PACF

Vzorec modelu SARIMA [29, 33] je na nasledujúcom Obrázku č. 7:

$$\nabla^d \nabla_S^D Y_t = \frac{\theta_q(B)\,\Theta_Q(B^S)}{\varphi_p(B)\,\Phi_P(B^S)}\varepsilon_t$$

$$\varphi_p(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots \varphi_p B^p$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots \theta_q B^q$$

$$\Phi_P(B^S) = 1 - \Phi_1 B^S - \phi_2 B^{2S} - \dots \phi_P B^{PS}$$

$$\Theta_Q(B^S) = 1 - \Theta_1 B^S - \Theta_2 B^{2S} - \dots \Theta_Q B^{QS}$$

Obr. 7: Zdroj [42].

kde parametre p, d, d, P, D, Q a S sme si bližšie priblížili pri Obrázku č. 4. Parameter B je operátor posunu vzad, Y_t predstavuje počet prípadov mumpsov v čase t a ε_t predstavuje odhadovaný zvyšok.

3.2 Prophet

Prophet je open source algoritmus vytvorený spoločnosťou Facebook. Cieľom tohto algoritmu je umožniť prognózovanie „v mierke“. Inými slovami, algoritmus chce byť nástrojom na predpovedanie, ktorý je prirodzene automatizovaný, čo zjednodušuje použitie pre ladenie metód časových radov. To umožňuje analytikom z rôznych prostredí a ľuďom s malými až žiadnymi predchádzajúcimi skúsenosťami v oblasti prognózovania úspešne predpovedať. Algoritmus sa vie veľmi dobre vysporiadať s chýbajúcimi údajmi, posunom trendu a s hodnotami, ktoré sú anomálie v datasete, ako napríklad príliš veľká hodnota alebo príliš malá hodnota. Prophet má svoj vlastný špeciálny dátový rámec na jednoduché spracovanie časových radov a sezónnosti. Dátový rámec potrebuje dva základné stĺpce. Jeden z týchto stĺpcov je *ds* a tento stĺpec ukladá časové radové dáta. Druhý stĺpec je y a ukladá súvisiace hodnoty časového radu v dátovom rámci. Týmto spôsobom môže rámec dobre fungovať na sezónnych časových radoch a poskytuje niektoré možnosti na spracovanie sezónnosti dátového súboru. Tieto možnosti sú ročná, týždenná a denná sezónnosť. Vďaka týmto možnostiam si analytik môže vybrať dostupnú časovú granularitu pre prognostický model na súbore údajov [45]. Podľa Facebooku Prophet najlepšie funguje s časovými radmi, ktoré majú silné sezónne efekty a niekoľko sezón historických údajov a je odolný voči odľahlým hodnotám a posunom v trende [37]. Tento algoritmus je založený na nasledujúcom vzorci:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t,$$

kde $y(t)$ označuje základný časový rad, ktorý sa má predpovedať, $g(t)$ zodpovedá za trendovú zložku zodpovednú za lineárne alebo nelineárne zmeny, $s(t)$ predstavuje periodickú časovú zložku v rôznych časových obdobiach, $h(t)$ reprezentuje prázdninové efekty na dáta a posledným prvkom vo vzorci je ε_t, ktorá predstavuje zvyškovú zložku [35].

3.3 SVR

Support vector regression (SVR) je metóda strojového učenia používaná na riešenie problémov regresie. Je to modifikácia metódy support vector machine (SVM), ktorá bola pôvodne vyvinutá pre riešenie problémov klasifikácie. Ide o metódu strojového učenia, ktorá využíva učiteľa. Cieľom SVR je naučiť sa funkciu, ktorá by čo najpresnejšie predikovala výstupy pre dané vstupy. Problémy regresie môžu byť lineárne alebo nelineárne a SVR bol vyvinutý najmä pre riešenie nelineárnych problémov regresie. Jedným z hlavných prvkov SVR je použitie kernel funkcií. Kernel funkcie umožňujú

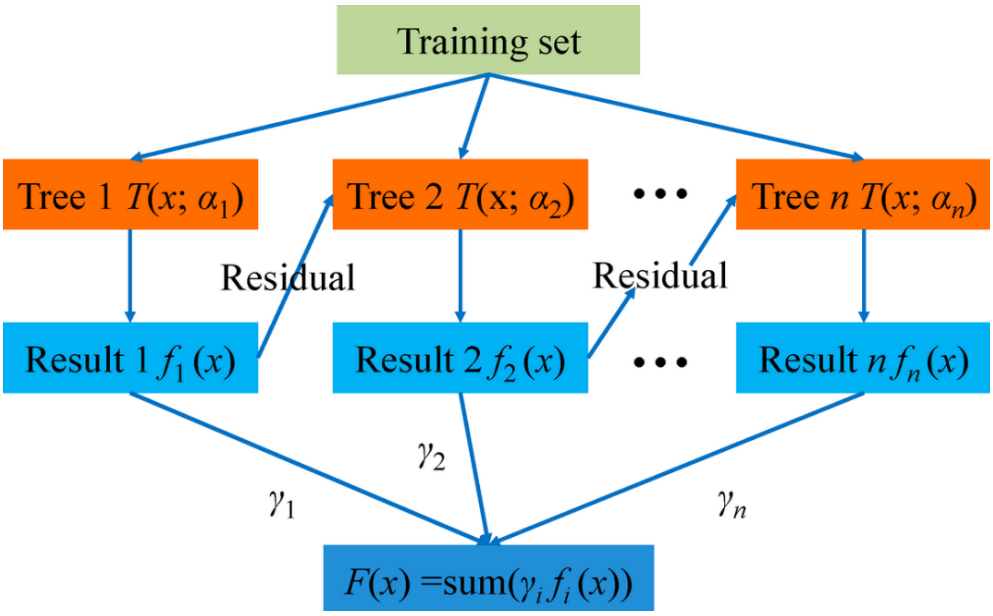
mapovať vstupné dáta do viac rozmerného priestoru, kde môžu byť lepšie lineárne korelované s výstupmi. To zvyšuje flexibilitu SVR a umožňuje hľadať riešenia v širšom rozsahu priestoru. Parametre SVR sa získavajú riešením kvadratického programovania s lineárnymi rovnostnými a nerovnostnými obmedzeniami [36]. Všeobecný vzorec odhadu funkcie [4] má túto podobu:

$$y = f(x) =< w, x > +b = \sum_{j=1}^{M} w_j x_j + b,$$

kde $y, b \in \mathbb{R}$ a $x, w \in \mathbb{R}^M$.

3.4 LightGBM

LightGBM je open source framework postavený na gradient boostingu, ktorý využíva stromové algoritmy. LightGBM trénuje pomocou skladania viacerých jednoduchých rozhodovacích stromov. Tento proces si vieme predstaviť pomocou Obrázka č. 8:



Obr. 8: Zdroj [28].

V práci [43] sa hovorí o tom, že každý strom sa učí rozdiel medzi skutočnými a predikovanými hodnotami z predchádzajúceho modelu a následne tieto stromy kombinuje. Tento algoritmus je vyjadrený týmto vzorcom:

$$F_M(x) = F_1(x) + \sum_{n=1}^{M} h_n(x)$$

$F_1(x)$ je prvý model, ktorý priamo predpovedá výslednú hodnotu. Rozhodovací strom rastie postupne v listoch, čo optimalizuje stratu, ktorá generuje vetvy [10]. Týmto spôsobom je tento algoritmus rýchlejší a menej komplexný ako rast na úrovni, ktorý rozširuje hĺbku stromu. Samotný algoritmus zlepšuje efektivitu a škálovateľnosť aj pri veľkom počte dát a atribútov, využitím dvoch nových techník: gradient-based one-side sampling a exclusive feature bundling. Prvá spomenutá technika prispieva k redukcii dát a to tak, že sú dáta vzorkované s integritou a gradientom a tým pádom sa model nemusí naučiť na všetkých dátach. Druhá technika prispieva k efektívnosti tým, že algoritmus nemusí prehľadávať všetky atribúty v dátach. Naopak, nezmyselné alebo nevýznamné atribúty, ktoré nemajú žiadny vplyv na predikciu, sú ignorované algoritmom. Tým sa znižuje počet atribútov, ktoré musí algoritmus analyzovať a zvyšuje sa rýchlosť trénovania a úspora pamäte. Použitie týchto techník umožňuje algoritmu výrazne prevýšiť podobné metódy, napr. XGBoost pokiaľ ide o rýchlosť učenia a úsporu pamäte [43]. Meidan a jeho tím v publikácii [30] stanovili, že časová zložitosť algoritmu sa vypočítava ako $\mathcal{O}(N_{data} \times M_{vlastnosti})$.

3.5 XGBoost

Model XGBoost bol prvýkrát navrhnutý Chenom Tianqim a Carlosom Gestrinom v roku 2011 a bol neustále optimalizovaný a zdokonaľovaný mnohými vedcami [3]. XGBoost je skratka pre Extreme Gradient Boosting a stal sa jedným z najpopulárnejších a najpoužívanejších algoritmov strojového učenia vďaka svojej schopnosti zvládnuť veľké súbory údajov a schopnosti dosahovať špičkový výkon v mnohých úlohách strojového učenia, ako je napr. klasifikácia a regresia. Algoritmus má svoje hlavné výhody v presnosti, flexibilite a automatickom spracovaní chýbajúcich hodnôt. Hlavnou myšlienkou XGBoostu je neustále pridávať k množine slabé stromy s rôznymi váhami [47]. V publikácii [3] je vzorec XGBoost modelu:

$$Obj_m = \sum_{i=1}^{n} l((y_i, y_i^{m-1}) + f_m(x_i)) + \Omega(f_m)$$

kde n predstavuje veľkosť vzorky, m predstavuje počet iterácií a f_m predstavuje chybu v m iteráciách l predstavuje stratovú funkciu, ktorá sa používa na meranie rozdielu medzi reálnou hodnotou a predpovedanou v poslednom kroku, ako aj na výstup nového stromu a Ω je regulačný termín, ktorý hodnotí zložitosť nového stromu [32]. Jednou z kľúčových vlastností XGBoost je jeho efektívne zaobchádzanie s chýbajúcimi hodnotami, čo mu umožňuje spracovávať reálne dáta s chýbajúcimi hodnotami bez toho, aby sa vyžadovalo značné predspracovanie. Okrem toho má XGBoost vstavanú podporu

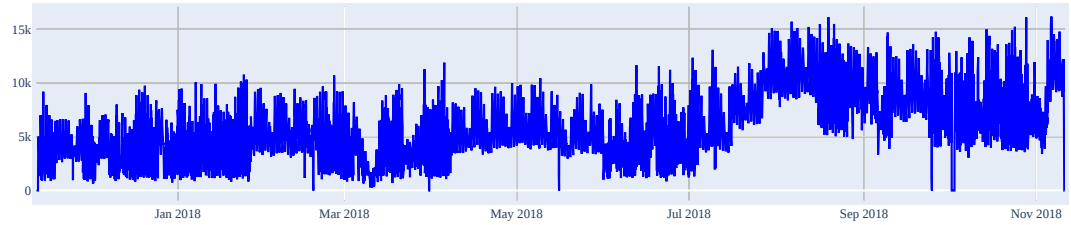
pre paralelné spracovanie, čo umožňuje trénovať modely na veľkých súboroch údajov v rozumnom čase.

4 Výsledky

V predchádzajúcich kapitolách sme sa venovali teoretickému základu vybraných metód strojového učenia a ich detailnému popisu. V tejto kapitole sa budeme podrobnejšie zaoberať ich implementáciou a výsledkami. Na vyhodnotenie presnosti predikcií, ktoré sme vytvorili pomocou jednotlivých metód strojového učenia, sme použili metriky MAE (Mean Absolute Error) a MASE (Mean Absolute Scaled Error), ktoré sú často používané v oblasti predikcie časových radov. Tieto metriky nám poskytli informáciu o tom, ako presné boli predikcie v porovnaní s reálnymi dátami. Okrem toho sme na porovnanie dvojíc predikcii použili aj Diebold-Mariano štatistický test. Zameriavali sme sa na jedno-krokovú predikciu časového radu, ktorý bol vytvorený pomocou počtu udalostí na sieťovom porte 445/TCP. Tento časový rad sme si zvolili z dôvodu jeho kvality dát v porovnaní s ostatnými časovými radmi. Použitá dátová sada je viac popísaná v nasledujúcej podkapitole. Jedným z našich cieľov bolo porovnať výsledky našich predikcií s výsledkami z existujúcich článkov, ktoré sa zameriavali na predikciu časových radov pomocou neurónových sieti a štatistických metód. To nám umožnilo zistiť, či sú naše výsledky v porovnaní s inými štúdiami relevantné a presné. Okrem toho popíšeme nastavenie jednotlivých parametrov modelov a ich optimalizáciu pre dosiahnutie čo najlepšej predikčnej presnosti. Tréning a validácia prebiehali na notebooku s procesorom AMD Ryzen 6800HS, 16GB RAM a operačnom systéme Windows 11.

4.1 Dátová sada

Dataset, ktorý používame v našej práci vznikol z českého systému WARDEN (systém pre zdieľanie informácií o detegovaných bezpečnostných udalostiach medzi organizáciami zapojenými do systému WARDEN [22]). Dataset pozostáva z 21 časových radov, ktoré boli vytvorené na základe rôznych kritérií. Napríklad Port 22, Port 443, Port 445, Category recon scanning, Protocol ICMP a podobne. Časové rady použité v tejto práci majú časovú jednotku 30 minút. Teda každý bod v časovom rade predstavuje počet bezpečnostných udalostí za posledných 30 minút pre konkrétne kritérium. Ukážka časového radu pre kritérium Port 445 je na obrázku 9.



Obr. 9: Ukážka časového radu vytvoreného podľa kritéria port 445/TCP(SMB) s časovou jednotkou 30 minút.

Práca [39] rozdeľuje túto dátovú sadu do troch skupín:

- **dobre predikovateľné časové rady** - zaraďujeme tu časový rad vytvorený podľa kritéria Port 445/TCP, ktorý je najlepšie predikovateľný v rámci tohoto datasetu.
- **nepredikovateľné časové rady** - obsahujúca 12 časových radov, ktorých predikcia je podobná naivnej predikcii. Obsahuje taktiež 2 podskupiny:
  - MASE hodnoty sa pohybujú pri jedno-krokovej predikcii v intervale 0,65-0,8 a pri desať krokovej sa pohybujú okolo hodnoty 1 alebo málo cez 1.
  - MASE hodnoty pohybujúce sa pri jedno-krokových predikciách nad 1 a viac pri desať krokových predikciách.
- **nepredikovateľný šum** - obsahuje rady vytvorené pomocou kategórií Attempt exploit a ICMP protokolu, ktoré majú veľmi blízko k bielemu šumu.

Aj vďaka predchádzajúcemu rozdeleniu sme sa rozhodli venovať našu pozornosť časovému radu, ktorý reprezentuje Port 445/TCP. Spomínaná práca mala k dispozícií dáta zo systému WARDEN z obdobia 11.12. 2017 až 11.12.2018. Dátová sada sa skladala z 17 473 bodov a z dôvodu chýbajúcich dát bola rozdelená na 3 časti. Prvá časť obsahovala hodnoty od 27 do 15 548. Druhá časť obsahovala hodnoty od 15 602 do 16 601 a tretia od 16 602 do 17 458. Časový rad reprezentujúci Port 445/TCP obsahuje celkovo 17473 záznamov a priemer týchto hodnôt je $5959,2$ so štandardnou odchýlkou $2912,71$. Minimálna hodnota v časovom rade je -1. Medián dosahuje hodnotu 5482. Maximálna hodnota v časovom rade je 16168. Tieto štatistiky poskytujú prehľad o vlastnostiach a charakteristikách časového radu, zahŕňajúc veľkosť, centrálne tendencie (priemer a medián) a variability (štandardná odchýlka).

4.2 Tréning a výsledky metód

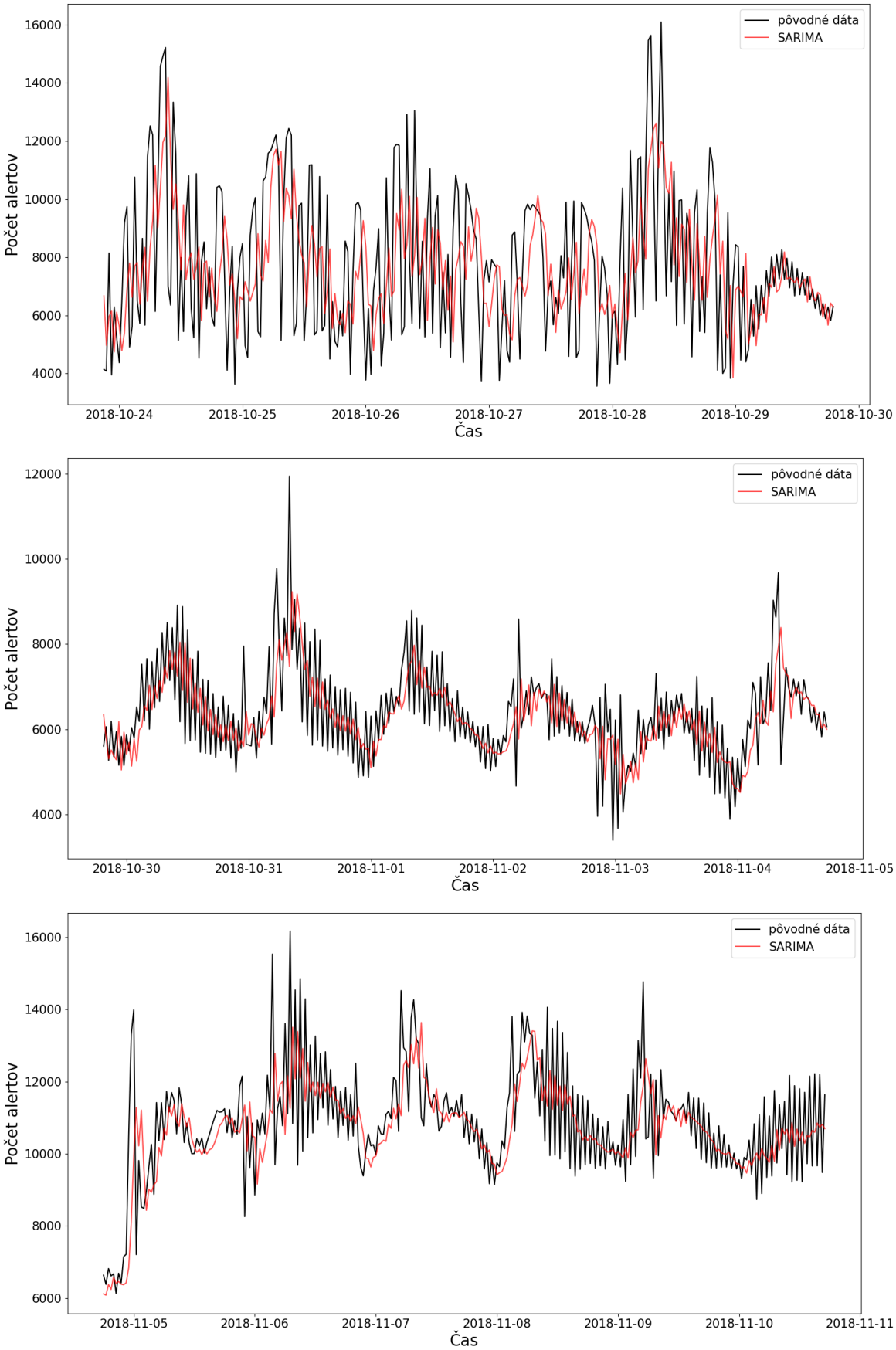
V tejto podkapitole popíšeme parametre a spôsob tréningu metód (viď. kapitola č.3). Všetky metódy boli implementované v jazyku Python. Na tréning modelov sme použili primárne druhú časť datasetu avšak pri trénovaní SVR modelu sme využili aj prvú časť datasetu. Posledná časť datasetu bola vyhradená ako testovacia množina.

SARIMA

Prvú metódu, ktorú sme implementovali bola SARIMA. Ďalšou dôležitou vlastnosťou časového radu je stacionarita. Na jej overenie sme použili rozšírený Dickey-Fuller test (Augmented Dickey-Fuller) test, ktorý nám umožnil overiť, či sú naše dáta stacionárne. Tento test sme vykonali pomocou tohto kódu:

```
result = adfuller(pd.concat((train1_df['30m-item71'],
val_df['30m-item71'])))
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
```

Výsledkom tohto testu je, že ADF štatistika je $-3,937608$ a p-hodnota je $0,001775$, čo naznačuje, že môžeme zamietnuť nulovú hypotézu o tom, že časový rad nie je stacionárny. To znamená, že naše dáta majú stacionárne štatistické vlastnosti a môžeme použiť metódy, ktoré predpokladajú stacionaritu. Kritické hodnoty pre 1%, 5% a 10% hladiny významnosti sú $-3,434$, $-2,863$ a $-2,568$, čo naznačuje, že s 99%, 95% a 90% istotou môžeme povedať, že náš časový rad je stacionárny. Pri implementácii tohto modelu sme najskôr vizuálne analyzovali ACF a PACF grafy s cieľom určiť vhodné parametre modelu. Pokúsili sme sa získať parametre modelu rôznymi spôsobmi, ale najpresnejšie výsledky nám poskytla knižnica auto-arima, ktorá pracuje na princípe grid searchu. Tento prístup nám vrátil optimálne parametre pre náš dataset, konkrétne $ARIMA(4,0,0)(0,0,1)_{48}$. Tieto parametre boli použité pri samotnej predikcii. Na trénovanie modelu sme použili hodnoty medzi 15602 a 16601, zatiaľ čo hodnoty od 16602 do 17458 boli vyhradené na testovanie modelu. Dáta neboli nijako upravované pre trénovanie a testovanie, pretože táto metóda je vhodná aj na jednorozmerné dáta. Tabuľka č. 3 poskytuje prehľad úspešnosti implementovaného modelu.



Obr. 10: Predikcia pomocou metódy SARIMA.

LightGBM a XGBoost

Ďalšími dvoma metódami, ktorým sme sa venovali boli algoritmy založené na gradientných rozhodovacích stromoch LightGBM a XGBoost. Pre tieto algoritmy je kľúčové, aby boli dáta uložené v dátovej štruktúre, ktorú môžu rýchlo a efektívne využívať ale tieto algoritmy nevedia predikovať, v prípade, že sa využíva jednorozmerné pole. Pri tejto implementácii by výsledkom bolo posunutie predikcie o jeden časový index, teda 30 minút. Vzhľadom na túto skutočnosť sme v dátach vytvorili posunutie pomocou lagov, ktoré vytvorilo ďalšie stĺpce v našich dátach. V následujúcom obrázku vidíme ukážku dát s použitím posunutia 1:

	30m-item71	30m-item71 + lag_1
2018-10-03 02:00:00	8668	5315.0
2018-10-03 02:30:00	5068	8668.0
2018-10-03 03:00:00	9559	5068.0
2018-10-03 03:30:00	10883	9559.0
2018-10-03 04:00:00	5656	10883.0
...	...	...
2018-11-10 15:00:00	12216	9662.0
2018-11-10 15:30:00	9666	12216.0
2018-11-10 16:00:00	12198	9666.0
2018-11-10 16:30:00	9482	12198.0
2018-11-10 17:00:00	11631	9482.0

Obr. 11: Ukážka dát pri posunutí 1

Ďalším krokom bolo transformovanie pôvodných dátových štruktúr do štruktúr určených pre tieto algoritmy. Napríklad LightGBM využíva efektívnejšiu dátovú štruktúru, ktorá sa nazýva lightgbm.Dataset a dokáže pracovať s veľkým objemom vstupných dát. Podobne ako LightGBM, aj XGBoost disponuje vlastnou dátovou štruktúrou, nazývanou xgboost.DMatrix. Jej využitie umožňuje algoritmu XGBoost rýchlo a efektívne spracovať dáta a dosiahnuť vysokú presnosť pri modelovaní. Nasledujúcim krokom bolo nájdenie najoptimálnejších parametrov pre tieto algoritmy. Pri trénovaní týchto algoritmov sme využili lagy s hodnotami 10, 25, 50 a 100. V oboch prípadoch sme použili 100 iterácií na trénovanie modelu. Pri hľadaní najvhodnejších parametrov sme využili niekoľko metód, vrátane grid search, knižnice optuna a aj implementácie modelov na základe empirického poznania a oficiálnej dokumentácie. Pri knižnici optuna sme pri hľadaní parametrov trénovali model na 150 iteráciách. Výsledkom bolo, že kombinácia empirického poznania s oficiálnou dokumentáciou sa osvedčila o čosi viac ako grid search alebo knižnica optuna pri hľadaní najoptimálnejších parametrov. Touto kombináciou sme získali nasledovné parametre pre LightGBM:

```
params = {
    "objective": "regression",
    "num_leaves": 32,
    "min_data_in_leaf": 10,
    "learning_rate": 0.05,
    "feature_fraction": 0.65,
    "bagging_fraction": 0.87,
    "seed": 1,
    "boosting_type": 'gbdt',
    'bagging_freq': 5,
    'verbose': 0,
    'max_depth': 5
}
```

Prostredníctvom knižnice optuna sme boli schopní nájsť parametre pri lagu 50:

```
optuna_params = {
    "objective": "regression",
    "num_leaves": 190,
    "min_data_in_leaf": 70,
    "learning_rate": 0.06704653473186421,
    "feature_fraction": 0.19911867675200498,
    "bagging_fraction": 0.5145569772992943,
    "boosting_type": 'gbdt',
    'bagging_freq': 2,
    'verbose': 0,
    'max_depth': 16
}
```

Tieto parametre boli postupne použité pri tréningu na dátach so všetkými lagmi.

Parameter `objective` určuje stratovú funkciu v našom prípade MSE a tým aj to, že riešime regresnú úlohu. Parameter `num_leaves` reprezentuje maximálny počet listov v jednom strome. Vyšší počet listov môže modelu umožniť zachytiť zložitejšie vzťahy v dátach, ale zároveň môže zvýšiť riziko pretrénovania. Ďalší parameter je `min_data_in_leaf`, ktorý určuje minimálny počet príkladov, ktoré sú vyžadované v každom liste rozhodovacieho stromu. Použitie tejto hodnoty pomáha zabrániť pretrénovaniu modelu a zlepšuje jeho schopnosť generalizovať na nové dáta. Parameter `learning_rate` nastavuje učiaci pomer pri použitý pri tréningu. Parameter `feature_fraction` sa používa na určenie podielu príznakov, ktoré sa náhodne vyberú pre

každú iteráciu rozhodovacieho stromu. Parameter `boosting_type` predstavuje typ boostovacieho algoritmu použitého v LightGBM. Bagging je technika výberu náhodnej vzorky (bez opakovania) zo vstupných dát pre každú iteráciu rozhodovacieho stromu, ktorá sa nastavuje parametrom `bagging_fraction`. Ďalším významným parametrom pri tréningu je `bagging_freq`, ktorý určuje frekvenciu, s akou sa aplikuje bagging. Parameter `max_depth` určuje maximálnu hĺbku rozhodovacieho stromu v algoritme. Hĺbka stromu zodpovedá počtu úrovní, ktoré sa v strome vytvárajú od koreňa po listy. Zvyšné parametre, ktoré nie sú spomenuté boli ponechané na predvolených hodnotách.

Pre model XGBoost sme určili tieto parametre:

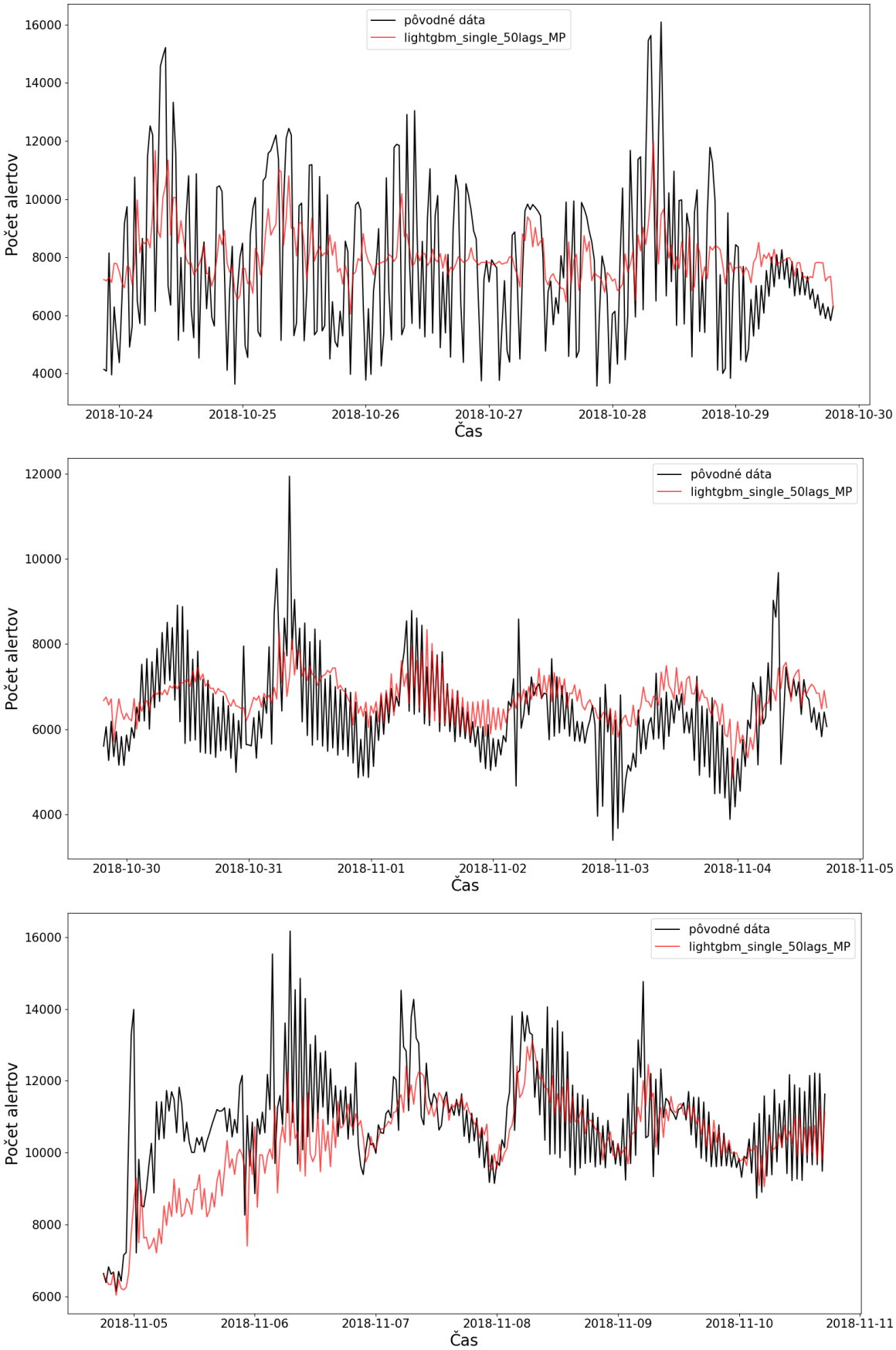
```
params = {
    "objective": "reg:squarederror",
    "max_depth": 6,
    "eta": 0.05,
    "subsample": 0.7,
    "colsample_bytree": 0.7,
    "seed": 1,
    "early_stopping_rounds": 10,
    "min_child_weight": 5
}
```

a spomínaná knižnica našla pre tento model pri použití lagu 50 tieto parametre:

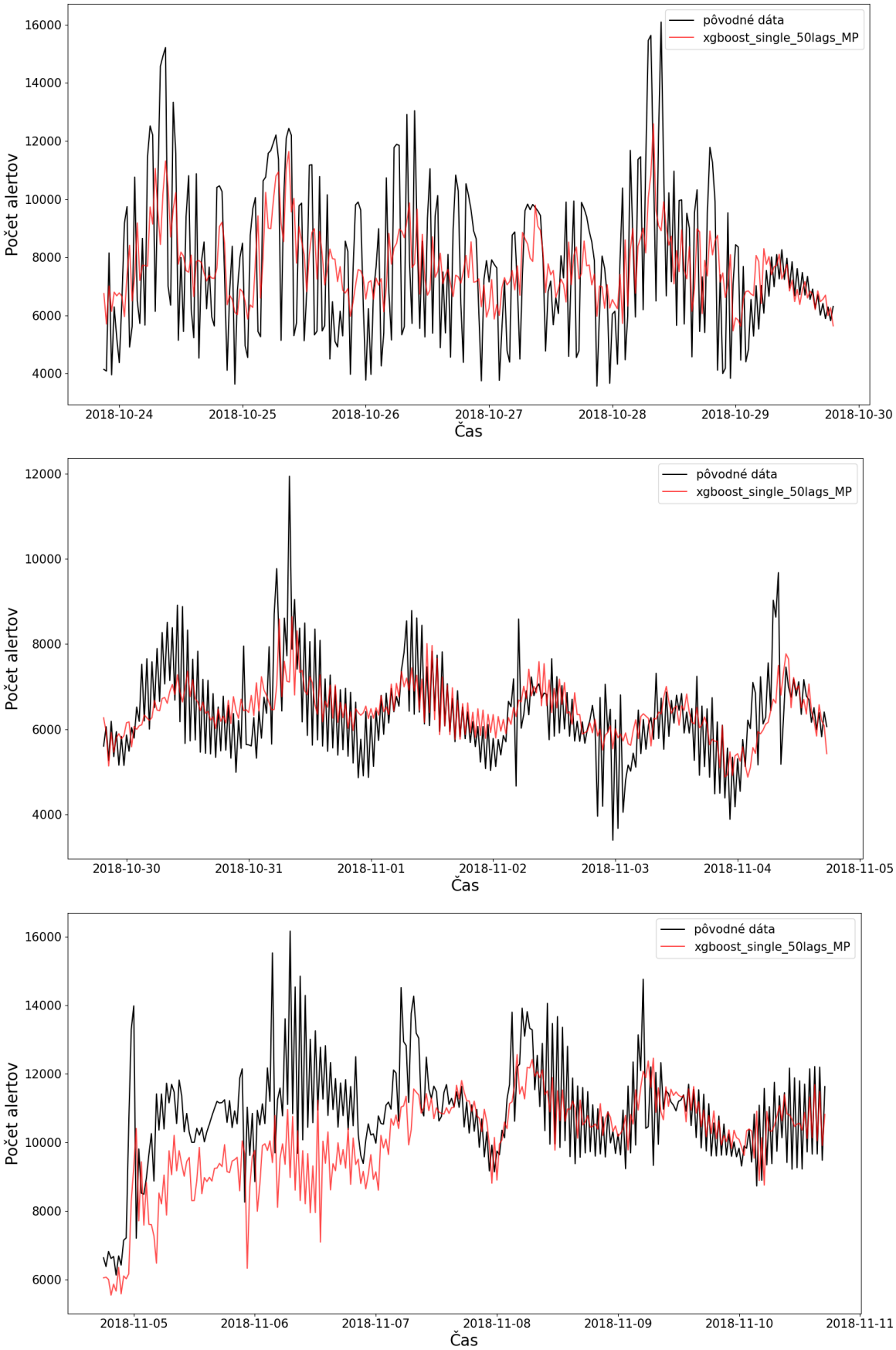
```
optuna_params = {
    'objective': 'reg:squarederror',
    'eval_metric': 'rmse',
    'booster': 'gbtree',
    'eta': 0.08458171122376003,
    'max_depth': 1,
    'subsample': 0.09101390864022472,
    'colsample_bytree': 0.3656323870928033,
    'min_child_weight': 17
}
```

Parameter `objective` s hodnotou reg:squarederror vyjadruje, že ide o regresnú úlohu so stratovou funkciou MSE. `Eval_metric` určuje metriku, ktorá sa používa na vyhodnocovanie modelu počas trénovania. V tomto prípade sme zvolili root mean square error (RMSE), čo je bežne používaná metrika pre regresné úlohy. Parameter `booster` s hodnotou gbtree určuje typ boostovacieho algoritmu použitého v algoritme. V tomto prípade je použitý rozhodovací strom (gradient boosting tree). `Eta`

je parameter, ktorý riadi veľkosť kroku (learning rate), ktorým sa aktualizujú váhy modelu po každej iterácii optimalizačného algoritmu. `Max_depth` určuje maximálnu hĺbku rozhodovacieho stromu v algoritme. `Subsample` určuje podiel náhodne vybraných vzoriek použitých pre každú iteráciu rozhodovacieho stromu. `Colsample_bytree` reprezentuje podiel príznakov, ktoré sa náhodne vyberú pre každý rozhodovací strom. `Min_child_weight` určuje minimálnu váhu príkladu potrebnú na vytvorenie ďalšej úrovne stromu. Zvyšné parametre, ktoré nie sú spomenuté, boli ponechané na predvolených hodnotách. S týmito množinami parametrov sme nakoniec pracovali pri vyhodnocovaní úspešnosti predikcie týchto algoritmov. Úspešnosť predikcii vzhľadom na počet lagov a výber parametrov si môžete pozrieť v Tabuľke č. 2.



Obr. 12: Predikcia pomocou metódy LightGBM.



Obr. 13: Predikcia pomocou metódy XGBoost.

Metódy	počet lagov	parametre	MAE	MASE
XGBoost	50	M	1291,3802	0,7591
XGBoost	25	M	1298,1773	0,7631
LightGBM	50	M	1301,8081	0,7652
LightGBM	100	O	1311,7483	0,7710
LightGBM	50	O	1317,2468	0,7743
XGBoost	25	O	1317,5040	0,7744
XGBoost	50	O	1318,9858	0,7753
LightGBM	25	M	1319,3209	0,7755
LightGBM	25	O	1327,4074	0,7802
XGBoost	10	M	1330,8707	0,7823
XGBoost	10	O	1337,1851	0,7860
LightGBM	10	M	1338,4631	0,7867
XGBoost	100	M	1342,6893	0,7892
XGBoost	100	O	1347,1369	0,7918
LightGBM	10	O	1356,0951	0,7971
LightGBM	100	M	1357,9942	0,7982

Tabuľka 2: Tabuľka zobrazuje vzostupne výsledky algoritmov založených na gradientných rozhodovacích stromoch vzhľadom na počet lagov a použitie parametrov. Symbol O predstavuje parametre určené knižnicou optuna a symbol M zasa parametre získané na základe empirických skúseností.

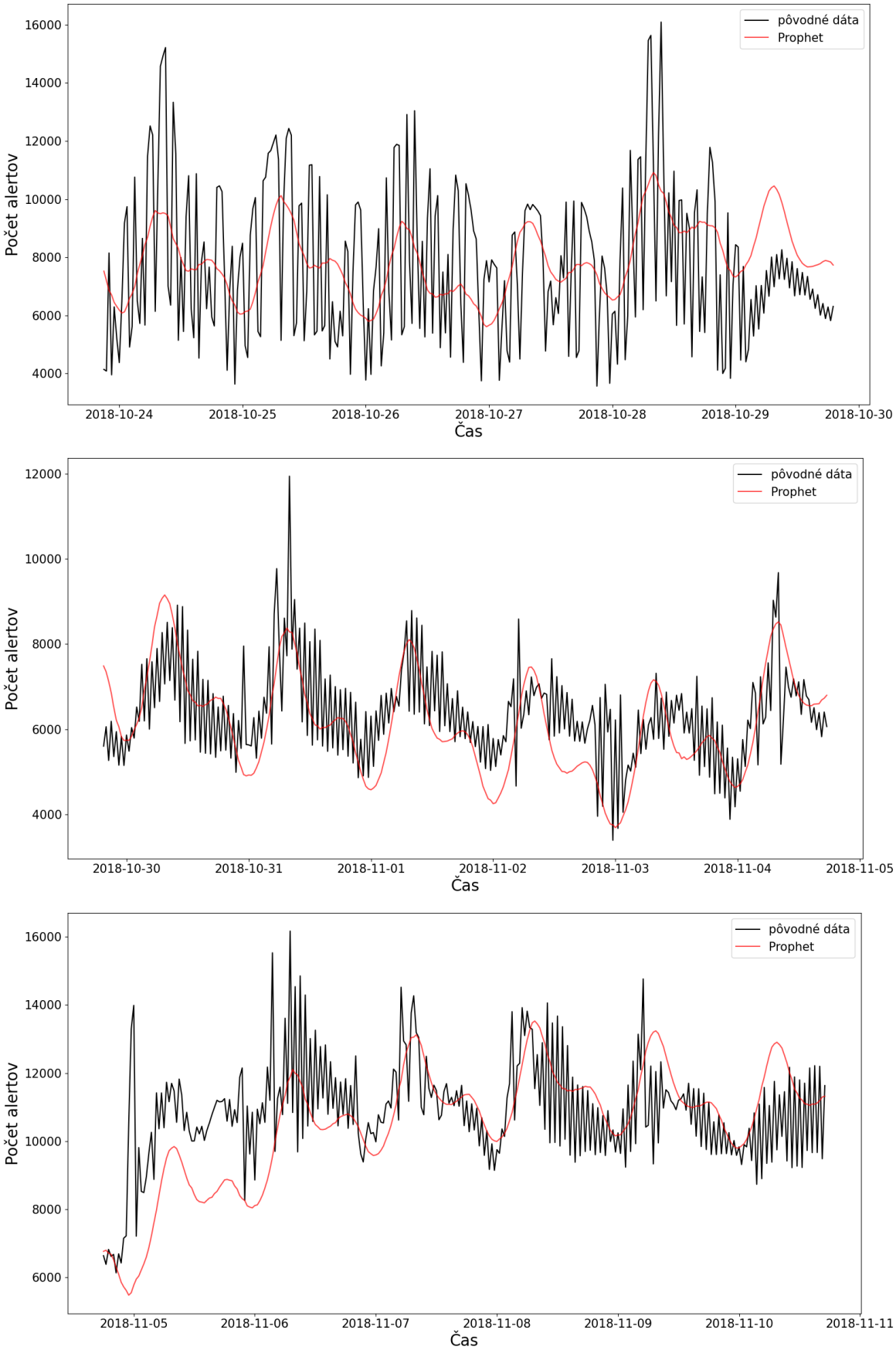
Prophet

Ďalšou metódou, ktorú sme implementovali bol algoritmus Prophet. Prvým krokom pri jeho implementácii bolo transformovať dáta do formátu, ktorý algoritmus očakáva. Konkrétne, algoritmus Prophet vyžaduje časový rad s dvoma stĺpcami *ds* pre časovú značku a y pre hodnotu, ktorá sa má predikovať. Po úprave dát sme pomocou grid search našli nasledovné vhodné parametre:

```
params = {
    'growth': 'linear',
    'seasonality_mode': 'additive',
    'changepoint_prior_scale': 0.5,
    'seasonality_prior_scale': 0.5,
    'interval_width': 0.95,
    'uncertainty_samples': 100
```

```
}
```

Parameter `growth` určuje typ rastu časového radu v algoritme. V tomto prípade je použitý lineárny rast. `Seasonality_mode` určuje typ sezónnosti v algoritme. V tomto prípade je použitá aditívna sezónnosť. Parameter `changepoint_prior_scale` určuje prioritu zmeny bodov v algoritme. Vyššia hodnota znamená väčšiu flexibilitu v identifikácii zmien. `Seasonality_prior_scale` určuje prioritu sezónnosti v algoritme. Vyššia hodnota znamená, že sezónnosť bude mať väčší vplyv na predikciu. Parameter `interval_width` určuje šírku intervalov spoľahlivosti v algoritme. V tomto prípade sme zvolili hodnotu 0,95, čo znamená, že budú vytvorené intervaly spoľahlivosti s 95% pokrytím. `Uncertainty_samples` určuje počet vzoriek použitých na odhad neistoty v algoritme. Vyšší počet vzoriek zvyčajne vedie k presnejšiemu odhadu neistoty. Zvyšné parametre, ktoré nie sú spomenuté, boli ponechané na predvolených hodnotách. Úspešnosť tohto algoritmu sú v Tabuľke č. 3.



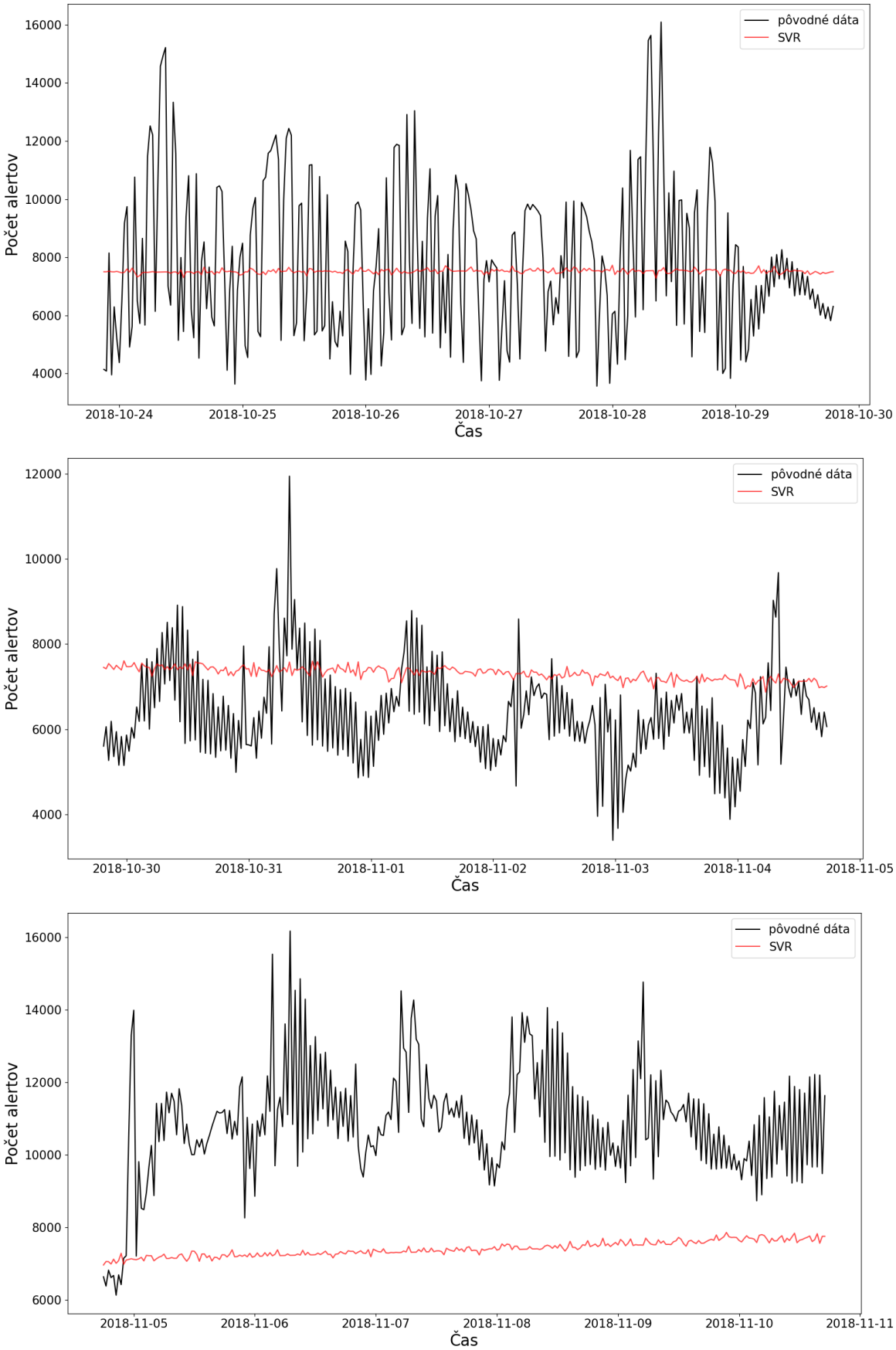
Obr. 14: Predikcia pomocou metódy Prophet.

SVR

Posledný algoritmus, ktorý sme implementovali bol SVR. Pri jeho implementácii sme rovnako ako pri ostatných algoritmoch použili grid search na nájdenie najlepších parametrov. Okrem toho sme pripravili vstupné dáta tak, aby obsahovali lag 1, teda hodnotu z predošlého časového kroku. Pri trénovaní modelu sme sa najprv použili druhú časť datasetu (hodnoty od 15602 do 16601). Avšak, výsledky na tejto časti boli veľmi nepresné v porovnaní s výsledkami iných metód a čas trénovania bol tiež podstatne väčší. Preto sme sa rozhodli natrénovať tento model na väčšom množstve dát, kde sme použili aj hodnoty od 27 až do 16601. Testovaciu množinu dát sme ponechali nezmenenú. Po trénovaní na väčšom množstve dát sme dosiahli výsledky MAE 2566,905808 a MASE 1,508796. Výsledok s použitím len druhej časti datasetu boli MAE 2271,270209 a MASE 1,335025. Ukážka najlepšej nami dosiahnutej predikcie je na obrázku 13.

Metódy	MAE	MASE
SARIMA	1306,8311	0,7681
XGBoost	**1291,3802**	**0,7591**
LightGBM	1301,8081	0,7652
Prophet	1434,5217	0,8432
SVR	2271,2702	1,3350

Tabuľka 3: Tabuľka najlepších výsledkov predikcii jednotlivých metód strojového učenia



Obr. 15: Predikcia pomocou metódy SVR.

Diebold-Mariano Test

Výsledky nami implementovaných modelov sme ešte vyhodnotili aplikovaním Diebold-Mariano štatistického testu. Nulová hypotéza spočívala v tom, že obe predikcie majú rovnakú presnosť a alternatívna hypotéza znamenala, že druhá metóda je menej presná ako prvá metóda. Stanovili sme, že p-hodnota pre potvrdenie nulovej hypotézy by mala byť vyššia ako 0,05 (zostávame pri 5% úrovni neistoty). Výsledky tohto testu sú v Tabuľke č. 4.

Metóda č.1	Metóda č.2	p-value
LightGBM	Prophet	**0,0073**
LightGBM	SARIMA	0,4837
LightGBM	SVR	**2,2e-16**
LightGBM	XGBoost	0,5702
Prophet	SARIMA	0,9737
Prophet	SVR	**2,2e-16**
Prophet	XGBoost	0,9863
Prophet	LightGBM	0,9927
SARIMA	Prophet	**0,0263**
SARIMA	LightGBM	0,5163
SARIMA	XGBoost	0,5539
SARIMA	SVR	**2,2e-16**
XGBoost	SARIMA	0,4461
XGBoost	Prophet	**0,01374**
XGBoost	LightGBM	0,4298
XGBoost	SVR	**2,2e-16**
SVR	SARIMA	1
SVR	Prophet	1
SVR	LightGBM	1
SVR	XGBoost	1

Tabuľka 4: Výsledky Diebold-Mariano testu pre jedno-krokovú predikciu metód strojového učenia.

Vzhľadom na výsledky v Tabuľke č. 4 pri hodnotách vyznačených hrubým písmom (menších ako 0,05) vieme povedať, že zamietame nulovú hypotézu a prijímame alternatívnu hypotézu. Teda napríklad v prvom riadku Tabuľky č. 4 je metóda Prophet menej presná ako LightGBM.

4.3 Porovnanie s existujúcimi výsledkami

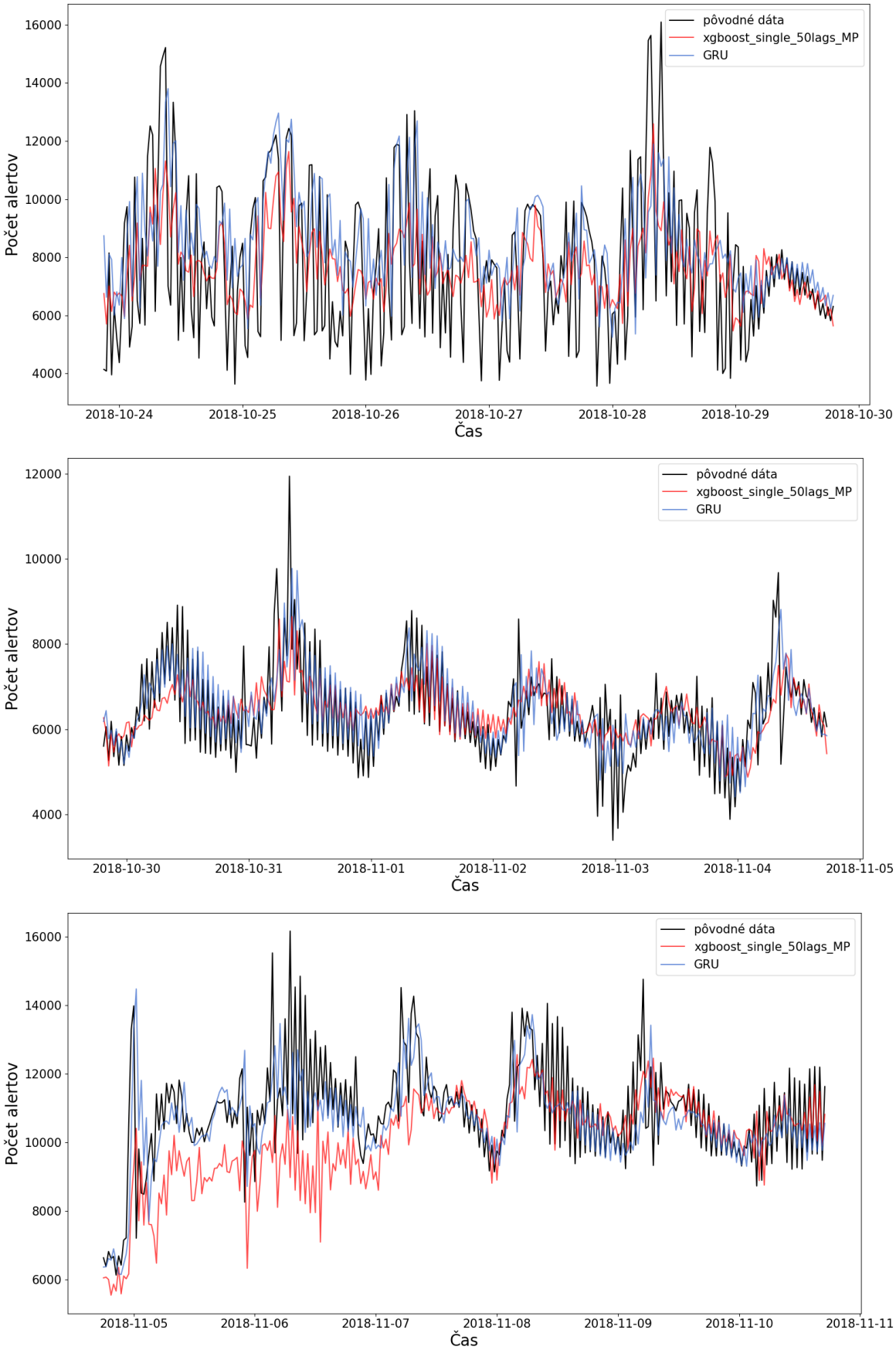
Posledným cieľom práce bolo porovnať výsledky s výsledkami iných prác. Naše výsledky sme porovnali s prácou [40], ktorá sa zaoberala rovnakou tematikou, využívajúcou štatistické metódy a neurónové siete. Číselne porovnanie sú v Tabuľke č. 5. Najlepšie výsledky boli dosiahnuté pri využití neurónových sieti, kedy sa hodnoty MAE pohybujú v intervale 1056,5102 - 1204,4519 a pre MASE 0,6210-0,7080, ktoré nebola schopná prekonať žiadna iná metóda. Naopak, najhoršie výsledky dosiahla predikcia prostredníctvom SVR metódy, ktoré sú v porovnaní s rekurentnou neurónovou sieťou typu GRU dvojnásobne hrošie. V porovnaní so štatistickými metódami sú naše výsledky získané pomocou algoritmov XGBoost a LightGBM lepšie iba minimálne.

Výsledky Diebold-Mariano štatistického testu urobeného na metódach tejto práce a metódach z práce [40] (rekurentná neurónová sieťou typu GRU a kombinácia metód ARIMA a Exponenciálne vyrovnávanie) sú v Tabuľke č. 6.

Vzhľadom na výsledky v Tabuľke č. 6 môžeme v prípadoch, kde je metóda č.1 rekurentná neurónová sieťou typu GRU tvrdiť, že zahadzujeme nulovú hypotézu a alternatívna hovori, že metóda č.2 je menej presná ako GRU. Teda metódy Prophet, LightGBM, XGBoost, SARIMA a SVR sú menej presné ako rekurentná neurónová sieťou typu GRU. Podobne v prípadoch AE - Prophet a AE - SVR, môžeme tvrdiť, že metódy Prophet a SVR sú horšie ako metóda AE.

Metódy	Typ	MAE	MASE
GRU	NN	1056,5102	0,6210
e1d1	NN	1090,1418	0,6408
LSTM	NN	1125,6894	0,6617
DN	NN	1186,1808	0,6972
Conv1D	NN	1204,4519	0,7080
XGBoost	ML	1291,3802	0,7591
LightGBM	ML	1301,8081	0,7652
E	S	1303,3641	0,7661
AE	S	1303,3641	0,7661
SARIMA	ML	1306,8311	0,7681
ARIMA	S	1352,5694	0,7950
Prophet	ML	1434,5217	0,8432
N	S	1686,0467	0,9910
SVR	ML	2271,2702	1,3350

Tabuľka 5: Porovnanie všetkých modelov vzostupne podľa metrík MAE a MASE. Poznámka: A - ARIMA, E - Exponenciálne vyrovnávanie; AE - ARIMA + Exponenciálne vyrovnávanie, N - naive model, GRU - rekurentná neurónová sieťou typu GRU, LSTM - rekurentná neurónová sieťou typu LSTM, e1d1 - neurónová sieť typu enkóder-dekóder, DN - dopredná neurónová sieť, Conv1D - 1D konvolučná neurónová sieť, ML - Strojové učenie, NN - Neurónová sieť, S - Štatistická metóda



Obr. 16: Grafické porovnanie najlepšieho modelu predikcie z neurnových sieti GRU a modelu strojového učenia XGBoost.

Metóda č.1	Metóda č.2	p-value
GRU	Prophet	**2,9565e-05**
GRU	LightGBM	**0,0009**
GRU	SVR	**9,8904e-44**
GRU	XGBoost	**0,0030**
GRU	SARIMA	**0,0006**
Prophet	GRU	1
LightGBM	GRU	0,9991
SVR	GRU	1,0
XGBoost	GRU	0,9970
SARIMA	GRU	0,9994
AE	Prophet	**0,0117**
AE	LightGBM	0,4784
AE	SVR	**7,3715e-39**
AE	XGBoost	0,5264
AE	SARIMA	0,4237
Prophet	AE	0,9883
LightGBM	AE	0,5216
SVR	AE	1
XGBoost	AE	0,4736
SARIMA	AE	0,5763

Tabuľka 6: Porovnanie výsledkov algoritmov tejto prác a rekurentnej neurónovej siete typu GRU a kombinácie metód ARIMA a Exponenciálne vyrovnávanie pomocou Diebold-Mariano testu. Poznámka: AE - ARIMA + Exponenciálne vyrovnávanie, GRU - rekurentná neurónová sieťou typu GRU

Záver

V tejto práci sme analyzovali existujúce prístupy využitia predikcie v sieťovom bezpečnostnom situačnom povedomí. Ukázalo sa, že metódy strojového učenia môžu byť aplikované pri predikcii časových radov v tejto oblasti. Úspešne sa nám podarilo implementovať všetky vybrané metódy na predikciu časových radov a v niektorých prípadoch dosahovali lepšie výsledky ako štatistické metódy použite v dostupnej literatúre. Najmä metódy XGBoost a LightGMB ukazujú sľubné výsledky, no ich použitie a použitie ďalších algoritmov založených na rozhodovacích stromoch si bude vyžadovať ďalšie preskúmanie. V rámci porovnania výsledkov zo strojového učenia s neurónovými sieťami sme zistili, že výsledky neurónových sietí boli lepšie a pre konkrétny dataset sú viac využiteľné. Výhodou nami implementovaných metód v tejto oblasti môže byt flexibilita nastavenia parametrov a odchyt sezónnosti pomocou konkrétnych metód v strojovom učení.

Vychádzajúc z výsledkov, ktoré boli výstupom skúmania, sa nám naskytajú ďalšie otázky, ktoré by mohli byť predmetom ďalšieho skúmania, ako napríklad využite iných metód strojového učenia alebo použitie iných druhov časových radov pripadne, zameranie sa na viac-krokovú predikciu.

Zoznam použitej literatúry

[1] Abdlhamed, M., Kifayat, K., Shi, Q., and Hurst, W. Intrusion prediction systems. In *Information fusion for cyber-security analytics*. Springer, 2017, pp. 155–174.

[2] Ahmed, A. A., and Zaman, N. A. K. Attack intention recognition: A review. *Int. J. Netw. Secur. 19*, 2 (2017), 244–250.

[3] Alim, M., Ye, G.-H., Guan, P., Huang, D.-S., Zhou, B.-S., and Wu, W. Comparison of arima model and xgboost model for prediction of human brucellosis in mainland china: a time-series study. *BMJ open 10*, 12 (2020), e039676.

[4] Awad, M., Khanna, R., Awad, M., and Khanna, R. Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers* (2015), 67–80.

[5] Bhuiyan, T. S. H., Choudhury, M. A. A. S., and Brooks, R. R. A framework for predictive cybersecurity intelligence. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)* (2017), vol. 2, IEEE, pp. 426–435.

[6] Brockwell, P. J., and Davis, R. A. *Introduction to Time Series and Forecasting*. Springer, 2016.

[7] Brownlee, J. A gentle introduction to sarima for time series forecasting in python [online]. Dostupné na internete: `https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/`, 2018. [cit. 11. 04. 2023].

[8] Chatfield, C. *The analysis of time series: an introduction*. CRC press, 2004.

[9] Chen, T., Chen, Y., Qian, C., Mao, J., and Liu, Y. Network anomaly detection based on autoregressive integrated moving average and support vector machine. *Journal of Internet Technology 16*, 5 (2015), 837–844.

[10] Chen, T., He, T., Benesty, M., Khotilovich, V., and Tang, Y. Xgboost—introduction to boosted trees. xgboost, 2017.

[11] El Naqa, I., and Murphy, M. J. *What Is Machine Learning?* Springer International Publishing, Cham, 2015, pp. 3–11.

[12] Endsley, M. R. Situation awareness global assessment technique (sagat). In *Proceedings of the IEEE 1988 national aerospace and electronics conference* (1988), IEEE, pp. 789–795.

[13] Endsley, M. R. Toward a theory of situation awareness in dynamic systems. *Human factors 37*, 1 (1995), 32–64.

[14] Geib, C. W., and Goldman, R. P. Plan recognition in intrusion detection systems. In *Proc. DARPA Inf. Survivability Conf. Exp. II (DISCEX)* (2001), vol. 1, pp. 46–55.

[15] Hughes, T., and Sheyner, O. Attack scenario graphs for computer network threat analysis and prediction. *Complexity 9*, 2 (2003), 15–18.

[16] Husák, M., Komárková, J., Bou-Harb, E., and Čeleda, P. Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys & Tutorials 21*, 1 (2018), 640–660.

[17] Hyndman, R. J., and Athanasopoulos, G. *Forecasting: principles and practice*. OTexts, 2018.

[18] Hyndman, R. J., and Athanasopoulos, G. *Forecasting: principles and practice, 3rd edition*. OTexts: Melbourne, Australia. OTexts.com/fpp3, 2021.

[19] Jakalan, A. Network security situational awareness. *The International Journal of Computer Science and Communication Security (IJCSCS) 3* (08 2013), 61–67.

[20] Jha, N. K., Ahn, G.-J., and Hu, H. Situational awareness in cyber security: The need for event aggregation and correlation. *IEEE Security & Privacy 9*, 6 (2011), 40–47.

[21] Jibao, L., Huiqiang, W., and Liang, Z. Study of network security situation awareness model based on simple additive weight and grey theory. In *2006 international conference on computational intelligence and security* (2006), vol. 2, IEEE, pp. 1545–1548.

[22] Kacha, P., Kostenec, M., and Kropacova, A. Warden 3: Security event exchange redesign. In *19th International Conference on Computers: Recent Advances in Computer Science* (2015).

[23] Kim, H. J., Cho, J. Y., Kim, D. K., and Choi, Y. J. Network security prediction using dynamic bayesian network and social network analysis. *Journal of Supercomputing 73*, 5 (2017), 2025–2041.

[24] Krizhevsky, D. D., and Hinton, G. Learning multiple layers of features from tiny images. Tech. rep., University of Toronto, 2009.

[25] Leau, Y.-B., and Manickam, S. Network security situation prediction: a review and discussion. In *International Conference on Soft Computing, Intelligence Systems, and Information Technology* (2015), Springer, pp. 424–435.

[26] Lee, J. H., Kim, J., and Kim, J. A survey of machine learning-based security threat detection techniques for iot devices. *Sensors 19*, 14 (2019), 3130.

[27] Lee, N.-U., Shim, J.-S., Ju, Y.-W., and Park, S.-C. Design and implementation of the sarima–svm time series analysis algorithm for the improvement of atmospheric environment forecast accuracy. *Soft Computing 22* (2018), 4275–4281.

[28] Liang, W., Luo, S., Zhao, G., and Wu, H. Predicting hard rock pillar stability using gbdt, xgboost, and lightgbm algorithms. *Mathematics 8*, 5 (2020), 765.

[29] Lin, Y., Chen, M., Chen, G., Wu, X., and Lin, T. Application of an autoregressive integrated moving average model for predicting injury mortality in xiamen, china. *BMJ open 5*, 12 (2015), e008491.

[30] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., and Elovici, Y. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing 17*, 3 (2018), 12–22.

[31] Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

[32] Nishio, M., Nishizawa, M., Sugiyama, O., Kojima, R., Yakami, M., Kuroda, T., and Togashi, K. Computer-aided diagnosis of lung nodule using gradient tree boosting and bayesian optimization. *PloS one 13*, 4 (2018), e0195875.

[33] Peng, Z., Bao, C., Zhao, Y., Yi, H., Tang, S., Yu, H., and Chen, F. Arima product season model and its application on forecasting in incidence of infectious disease. *Appl Stat Management 27*, 2 (2008), 362–368.

[34] Qin, X., and Lee, W. Attack plan recognition and prediction using causal networks. In *20th Annual Computer Security Applications Conference* (2004), IEEE, pp. 370–379.

[35] Samal, K. K. R., Babu, K. S., Das, S. K., and Acharaya, A. Time series based air pollution forecasting using sarima and prophet model. In *proceedings of the 2019 international conference on information technology and computer communications* (2019), pp. 80–85.

[36] Santos, L. L., and Castillo, F. S. Introduction to spatial network forecast with r [online]. Dostupné na internete: `https://laurentlsantos.github.io/forecasting/support-vector-regression.html`, 2019. [cit. 11. 04. 2023].

[37] Satrio, C. B. A., Darmawan, W., Nadia, B. U., and Hanafiah, N. Time series analysis and forecasting of coronavirus disease in indonesia using arima model and prophet. *Procedia Computer Science 179* (2021), 524–532.

[38] Shumway, R. H., and Stoffer, D. S. *Time series analysis and its applications: with R examples*. Springer, 2017.

[39] Sokol, P., Staňa, R., Gajdoš, A., and Pekarčík, P. Network security situation awareness forecasting based on statistical approach and neural networks. *Logic Journal of the IGPL* (2022).

[40] Staňa, R., Pekarčík, P., Gajdoš, A., and Sokol, P. Network security situation awareness forecasting based on neural networks. In *Theory and Applications of Time Series Analysis and Forecasting: Selected Contributions from ITISE 2021*. Springer, 2022, pp. 255–270.

[41] Tsay, R. S. *Analysis of Financial Time Series*. John Wiley Sons, 2018.

[42] Xu, Q., Li, R., Liu, Y., Luo, C., Xu, A., Xue, F., Xu, Q., and Li, X. Forecasting the incidence of mumps in zibo city based on a sarima model. *International journal of environmental research and public health 14*, 8 (2017), 925.

[43] Yanabe, T., Nishi, H., and Hashimoto, M. Anomaly detection based on histogram methodology and factor analysis using lightgbm for cooling systems. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (2020), vol. 1, pp. 952–958.

[44] Yang, S. J., Du, H., Holsopple, J., and Sudit, M. Attack projection. *Cyber Defense and Situational Awareness* (2014), 239–261.

[45] Yenidoğan, I., Çayir, A., Kozan, O., Dağ, T., and Arslan, Ç. Bitcoin forecasting using arima and prophet. In *2018 3rd international conference on computer science and engineering (UBMK)* (2018), IEEE, pp. 621–624.

[46] Yuan, Y., and Wang, Q. Anomaly-based network intrusion detection by mining software behaviors. *Information Sciences 339* (2016), 13–28.

[47] Zhang, L., Bian, W., Qu, W., Tuo, L., and Wang, Y. Time series forecast of sales volume based on xgboost. In *Journal of Physics: Conference Series* (2021), vol. 1873, IOP Publishing, p. 012067.

[48] Zhang, Y., Sun, Y., Zhang, L., Wang, K., and Li, J. An attack projection method based on deep learning for targeted malware. *IEEE Access 8* (2020), 167448–167458.

Prílohy

Príloha A: Všetky zdrojové kódy, ktoré boli použite pri tvorbe tejto práce sú dostupne na https://gitlab.science.upjs.sk/AlexGajdos/predikcia-ml