UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

VYUŽITIE METÓD STROJOVÉHO UČENIA V ANALÝZE BEZPEČNOSTNÝCH ÚDAJOV

Rigorózna práca

2022 **Mgr. Richard Staňa**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

VYUŽITIE METÓD STROJOVÉHO UČENIA V ANALÝZE BEZPEČNOSTNÝCH ÚDAJOV

Rigorózna práca

Študijný odbor:	Informatika
Školiace pracovisko:	Ústav informatiky
Vedúci práce:	prof. RNDr. Gabriel Semanišin, PhD.
Konzultant:	RNDr. JUDr. Pavol Sokol, PhD.

Košice 2022 **Mgr. Richard Staňa**

Prírodovedecká fakulta Univerzity P.J.Šafárika v Košiciach

Z A D A N I E
témy rigoróznej práce a predmetov rigoróznej skúšky

Meno a priezvisko: Mgr. Richard Staňa

Študijný odbor: Informatika

Konzultant: Prof. RNDr. Gabriel Semanišin, PhD.

Téma rigoróznej práce:
Využitie metód strojového učenia v analýze bezpečnostných údajov

Cieľ práce:
1. Analýza existujúcich aplikácii strojového učenia v oblasti počítačovej bezpečnosti.
2. Porovnanie výsledkov známych metód na existujúcich dátových sadách pre vybraný problém.
3. Vytváranie dátových sád o bezpečnostných incidentoch z reálnej sieťovej prevádzky a ich následná analýza.
4. Analýza vytvorených dát a porovnanie získaných výsledkov s alternatívnymi prístupmi.

Odporúčaná literatúra:
[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning. MIT Press, 2016.
[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv 1312.6199, 2013.
[3] Ross Girshick. Fast R-CNN. In ICCV, 2015.
[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. TPAMI, 2017.
[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In CVPR, pages 779–788, 2016.
[6] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

Termín odovzdania práce: 30.8.2022
Predmety rigoróznej skúšky: 1. Algoritmy a štruktúry dát
2. Neurónové siete

V Košiciach, dňa: ----------
Prof. RNDr. Stanislav Krajči, PhD.

Abstrakt

Každý administrátor by mal mať prehľad o bezpečnostnom stave siete, ktorú spravuje a vedieť približne odhadnúť ako sa bude vyvíjať v budúcnosti. Tato úloha je však veľmi náročná z viacerých dôvodov ako sú nedostatok dát, náročnosť ich pochopenia a podobne. Cieľom tejto práce je zlepšiť predikciu sieťového bezpečnostného situačného povedomia. Chceme tak urobiť pomocou vytvorenia kvalitnej dátovej sady časových radov, ktorý by pomohol komunite v ďalšom výskume v tejto oblasti. Okrem toho sa chceme pozrieť akými spôsobmi je možné zlepšiť samotnú predikciu časových radov v oblasti sieťového bezpečnostného situačného povedomia, na základe dát samotných (použitím externých údajov a augumentácie dát) a na základe práce s metódami (použitím skladania modelov a neštandardných loss funkcií).

Kľúčové slová: *sieťové bezpečnostné situačné povedomie, dátová sada, predikcia časových radov.*

Abstract

Every administrator should be aware of cybersecurity situation status of network and he should be able to estimate how it will evolve in future. This task is very demanding due to several reasons such as lack of data, difficulty in theyr understanding and so on. The aim of this thesis to improve the prediction of network security situational awareness. We want to accomplish that by creating high-quality time series dataset that would help the community in further research in this area. In addition, we want find other posibillities how to improve time series forecasting in the area of network security situational awareness, based on the data itself (using external data and data augmentation) and improvement of methods (using model ensembling and non-standard loss functions).

Keywords: *network cybersecurity situation awareness, dataset, time series forecasting.*

Obsah

Úvod

Útoky na počítačové siete prebiehajú neustále a žiadnu sieť nie je možné zabezpečiť dokonale. S neustále rastúcou internetovou prevádzkou rastie aj sila a množstvo prebiehajúcich útokov a bezpečnostných incidentov. Organizácie musia vyvíjať stále väčšie úsilie na to, aby sa chránili pred kybernetickými hrozbami. Obrana pred kybernetickými hrozbami je však finančne a časovo náročná. V prípade malých organizácií je možné zabezpečovať dostatočnú bezpečnosť jednoduchými nástrojmi (ako antimalvérové riešenia) a niekoľkými správcami, ktorí riešia aj bezpečnostné incidenty. So zväčšujúcou sa organizáciou nebudú ale jednoduché nástroje a neškolený neprofesionálny personál stačiť. Na rad budú prichádzať školení bezpečnostní analytici, ktorí vedia posúdiť závažnosti vzniknutých bezpečnostných incidentov, ale aj všeobecnú situáciu v počítačovej sieti. Okrem toho budú samozrejmosťou špecializované nástroje, ktoré budú týmto analytikom pomáhať zhromažďovať údaje, či už o vzniknutých incidentoch alebo o aktuálnej situácií v počítačovej sieti a pomôžu im predikovať budúci vývoj tejto situácie a tak im pomôžu zabezpečiť kybernetickú bezpečnosť organizácie.

Týmito nástrojmi sú často honeypoty, systémy na detekciu prienikov (IDS), systémy na prevenciu útokov (IPS) a rôzne systémy slúžiace na zaznamenávanie udalostí. Údaje z týchto systémov sú veľmi dôležitým zdrojom informácií o dianí v rámci počítačovej siete organizácie, ktorú môžeme označiť ako povedomie o sieťovom bezpečnostnom povedomí. Častým prístupom k spracovaniu údajov zo spomínaných systémov je vytváranie časových radov na základe počtu udalostí za jednotku času. Tieto časové rady sú následne používané na rôzne účely, napríklad detekciu anomálií alebo predikciu. Takáto predikcia sieťového bezpečnostného situačného povedomia je najvšeobecnejšia oblasť predikcie v kybernetickej bezpečnosti.

V rámci tejto práce sa zameriavame práve na sieťové bezpečnostné situačné povedomie. Je všeobecne známe, že metódy strojového učenia nedosahujú požadované výsledky bez kvalitnej a dostatočne veľkej dátovej sady. Žiaľ v oblasti, ktorej sa chceme venovať, sme nenašli dátovú sadu, na ktorej by sme mohli vyskúšať kvalitu nami testovaných algoritmov. Našou snahou je prispieť komunite venujúcej sa kybernetickej bezpečnosti, vytvorením kvalitnej dátovej sady, ktorá v tejto oblasti chýba. Takáto

dátová sada by výskumníkom v danej oblasti pomohla dostatočne overiť kvalitu nimi predstavovaných algoritmov na predikciu sieťového bezpečnostného situačného povedomia. Okrem toho sa chceme zamerať aj na samotnú predikciu sieťového bezpečnostného situačného povedomia. Ukazuje sa, že publikované práce málo pracujú s dátami, ktoré majú k dispozícií. Snažíme sa teda zlepšiť kvalitu predikcie sieťového bezpečnostného situačného povedomia pomocou použitia externých údajov a pomocou augumentácie dát. Ďalším problémom, ktorý vnímame, je to, že existujúce výskumy používajú iba jednoduché metódy predikcie časových radov. Zameriavame sa preto na rôzne moderné metódy strojového učenia, neurónové siete a metódy skladania modelov a chceme overiť vplyv použitia neštandardných stratových funkcií pri tréningu neurónových sietí na kvalitu predikcie.

Práca sa skladá zo štyroch kapitol. V prvých dvoch sú uvedené teoretické poznatky z predikcie v oblasti kybernetickej bezpečnosti a zo samotnej predikcie časových radov. Tretia popisuje výskumné ciele, ktorým sa venujeme s aktuálnym stavom riešenej problematiky a nami zvolenými prístupmi pre riešenie danej problematiky. V poslednej kapitole sú uvedené nami dosiahnuté výsledky, ktoré sme publikovali vo výskumných článkoch a čiastočne riešenia nami stanovených výskumných cieľov.

1 Predikcia v oblasti kybernetickej bezpečnosti

Podľa prehľadového článku [36] vieme rozdeliť úlohu predikcie v oblasti kybernetickej bezpečnosti na tri prípady. Ich sumár nájdeme v tabuľke 1. Historicky je prvý prípad **projekcia útoku**, respektíve rozpoznanie zámeru útočníka. Úlohou je predpovedať, aký je ďalší krok útočníka už v prebiehajúcom útoku, respektíve aký je finálny cieľ útočníka. Druhým prípadom je **predikcia útoku**. Hlavnou úlohou je predikovať útok pred tým, ako nastane. Posledným prípad je **predikcia sieťovej bezpečnostnej situácie**, ktorý je veľmi všeobecný a súvisí s kybernetickým situačným povedomím. Úlohou je v tomto prípade predikovať situáciu v celej sieti. V nasledujúcej časti si jednotlivé prípady popíšeme konkrétnejšie podobne ako sú popísane v práci [36].

1.1 Projekcia útokov

Prvé metódy projekcie útoku sa začali objavovať okolo roku 2003 v prácach [35, 69]. Oblasť výskumu je stále aktívna a nájdeme ju aj v prehľadových článkoch [102, 2]. Na to, aby sme mohli popísať kroky útoku a pokúsiť sa ich predikovať, potrebujeme najprv zdokumentovať štandardné správanie útočníkov a vytvoriť popis krokov útoku. Príklad takejto postupnosti krokov útoku nájdeme v práci [8]:

prípad	úloha	ďalší prehľad
projekcia útoku/ rozpoznanie zámeru útočníka	predpovedať ďalší krok útočníka/ finálny cieľ útočníka	[102, 2]
predikcia útoku	predpovedať kedy, kde a aký typ útoku nastane	[1]
predikcia sieťovej bezpečnostnej situácie	predpovedať ako sa vyvinie celková situácia v sieti	[47]

Tabuľka 1: Tabuľka oblastí predikcie v kybernetickej bezpečnosti. [36]

1. skenovanie
2. enumerácia
3. pokus o prienik
4. zvýšenie oprávnení
5. vykonanie škodlivých činností
6. nasadenie malvéru/zadných vrátok
7. odstránenie evidencie a ukončenie

Mnoho typov útokov používa tieto jednoduché kroky a je možné ich sledovať či už v sieťovej prevádzke alebo na cieľovom systéme. Projekcia útoku je v základe veľmi jednoduchá. Ak vidíme sekvenciu krokov, ktorá je známa, môžeme predpokladať, ako bude útočník postupovať podľa známych krokov. Takýto vágny popis útoku žiaľ nie je použiteľný pre algoritmickú predikciu, a teda je potrebný viac formálny popis útoku, štandardne pomocou grafov útoku [35]. Existuje mnoho rôznych typov útoku, takže je potrebné vytvoriť model, ktorý bude môcť byť použitý na projekciu viacerých útokov. Historicky prvé modely boli vytvárané ručne. Z tohto dôvodu ich bolo potrebné aj ručne aktualizovať. Moderné metódy sú často založené na dolovaní dát, ktoré automaticky generujú vzory útoku pre projekciu útokov.

V prípade rozpoznávania zámeru útočníka je idea veľmi podobná, rozdiel je v motivácií. Ak odhadneme cieľ útočníka, môžeme predpokladať budúce bezpečnostné udalosti podľa konkrétneho útoku. Rozpoznávanie zámeru útočníka nájdeme v práci [2], kde sa sústredili na historické dáta. Nové techniky sa snažia zameriavať na rozpoznávanie zámeru útočníka v reálnom čase a stále viac sa približujú projekcii útoku.

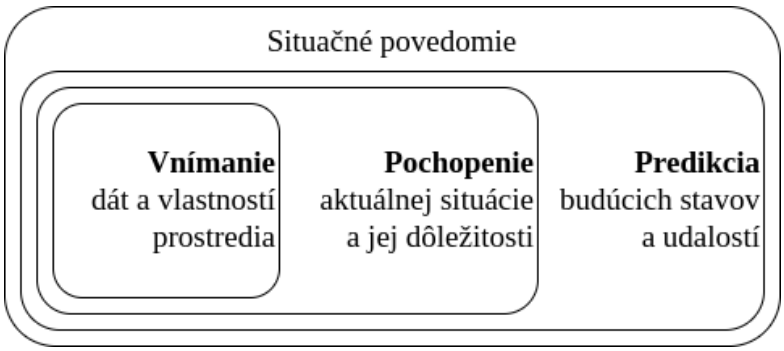
1.2 Predikcia útokov

Viac všeobecná úloha je úloha predikcie útokov, najmä prienikov do systému [1]. Namiesto projekcie prebiehajúceho útoku ide o predikciu nového útoku. Vzhľadom na príliš všeobecnú povahu úlohy nie je v existujúcich prístupoch veľa spoločných prvkov. Kým metódy projekcie útokov sa spoliehajú najmä na diskrétne modely kybernetických útokov, metódy a modely používané pre predikciu útoku sú rôzne, od diskrétnych ako graf útokov až po spojité ako napr. časové rady. Predikcia útoku je možná pomocou rovnakého diskrétneho modelu použitého pre projekciu útoku s obmenou na začiatku. Napríklad predikcia nezačne s aktuálne prebiehajúcou škodlivou udalosťou ale s pravdepodobnosťou, že bude objavená konkrétna zraniteľnosť v počítačovej sieti. Prístup

spojitého modelu založeného na časových radoch počtu útokov na konkrétny systém alebo sieť v čase, môže byť použitý na predikciu toho, či útok nastane alebo nie. Pokročilé metódy môžu pracovať s typmi útoku a charakteristikami útočníka a obete a tak môžu odhadnúť, aký typ útoku nastane, kto bude útočníkom a kto bude cieľom útoku. Nové metódy často používajú aj iné zdroje dát na predikciu, napríklad informácie zo sociálnych sietí alebo sledujú zmeny v správaní používateľov.

1.3 Predikcia sieťovej bezpečnostnej situácie

Posledným prípadom predikcie v kybernetickej bezpečnosti je predikcia sieťovej bezpečnostnej situácie. Úloha je v tomto prípade ešte všeobecnejšia, pretože sa nezameriava na individuálneho útočníka alebo na prebiehajúci útok, ale na aktuálny globálny stav systému alebo siete pod našou kontrolou. Tento prípad je viac rozoberaný v práci [47]. Kľúčový koncept tohoto pohľadu na kybernetickú bezpečnosť sa často označuje ako kybernetické situačné povedomie (CSA - cyber situational awareness) alebo ako **sieťové bezpečnostné situačné povedomie** (NSSA - network security situational awareness). V práci sa budeme venovať NSSA. Oba pojmy majú pôvod v pojme situačné povedomie, ktorý vznikol vo vojenskom výskume. Najširšie používaná definícia situačného povedomia je z práce [23]: *Vnímanie prvkov v prostredí v čase a priestore, pochopenie ich významu a predikcia ich hodnôt v blízkej budúcnosti.* Definícia popisuje 3 stupne situačného povedomia *vnímanie*, *pochopenie*, *predikcia* tak ako je popísané na obrázku 1.



Obr. 1: Stupne situačného povedomia. Zdroj [24].

Keď tieto pojmy aplikujeme v oblasti kybernetickej bezpečnosti, tak vnímanie zodpovedá monitorovaniu kybernetických systémov, ale aj detekcii prienikov, pochopenie zodpovedá pochopeniu kybernetickej situácie, teda modelovaniu kybernetických hrozieb alebo korelovaniu bezpečnostných upozornení a predikcia hovorí o predikovaní

zmien v kybernetickej bezpečnostnej situácii. Dôležitosť predikcie je hlboko zakorenená v teoretickom pozadí situačného povedomia, a preto motivuje výskum predikcie v oblasti kybernetickej bezpečnosti.

Väčšina prác používa kvantitatívnu analýzu na popis sieťovej bezpečnostnej situácie v určitom čase [36]. Výsledné hodnoty sú následne použité na predpovedanie budúcej situácie. Takýto prístup nedáva žiadnu pridanú informáciu o presnej povahe budúcich útokov. Môže však 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. Žiaľ neexistuje žiadna zavedená a štandardne používaná metrika. Existujú však dva prístupy: hierarchická metóda s aditívnymi váhami a metóda odhadu intenzity útoku.

Hierarchická metóda vyhodnocuje sieťovú situáciu zdola hore. Najprv sa bezpečnostná situácia vyhodnotí pre každé koncové zariadenie. Následne je hodnota každého zariadenia vynásobená prislúchajúcou váhou a hodnoty sú spočítané, pre získanie celkovej bezpečnostnej sieťovej situácie siete. Rôzni autori používajú rôzne metódy na vyhodnotenie bezpečnostnej situácie koncových zariadení. Váhy väčšinou väčšinou vyjadrujú dôležitosť zariadení.

Metóda odhadu intenzity útoku spája informácie o prebiehajúcich útokoch z rôznych zdrojov a odhaduje celkovú intenzitu útoku. Celková intenzita je odvodená z počtu a závažnosti útokov proti celej sieti. Predikcie potom môžu poskytnúť varovanie o prichádzajúcom náraste alebo poklese útokov. Keďže vstup aj predpovedaná hodnota sú numerické hodnoty, väčšina modelov použitých na predikciu sieťovej bezpečnostnej situácie spadá do kategórie spojitých.

2 Predikcia časových radov

Ako sme popísali v predchádzajúcej časti, v oblasti kybernetickej bezpečnosti vieme nájsť viacero prípadov na predikciu. V našej práci sa budeme venovať najmä predpovedaniu sieťovej bezpečnostnej situácie pomocou časových radov. V nasledujúcom texte inšpirovanom [37] si priblížime predikciu časových radov ako takých.

Časový rad je séria hodnôt (informácií, dát a pod.), ktoré sú indexované s časovým usporiadaním. Najčastejšie je to sekvencia hodnôt, ktoré boli zozbierané s rovnakým časovým odstupom. Väčšinou sú to buď namerané hodnoty (teplota vzduchu, ceny na burze a pod.) alebo počet výskytov udalostí za stanovenú jednotku času (počet áut, ktoré prešli cez v oboch smeroch za hodinu, počet pokusov o prihlásenie na SSH server za posledných 10 minút a pod.)

Predikcia je potrebná v rôznych situáciách. Napríklad, bude spotreba elektrickej energie v najbližších piatich rokoch tak vysoká, že bude potrebné postaviť ďalšiu elektráreň? Ako správne rozvrhnúť počet zamestnancov v call centre budúci týždeň podľa počtu prichádzajúcich hovorov? Ako správne upozorniť administrátora siete na prichádzajúcu hrozbu útoku podľa dát z rôznych bezpečnostných sond? Niektoré veci sa dajú predikovať ľahšie a niektoré takmer vôbec. Napríklad čas zajtrajšieho východu a západu slnka vieme odhadnúť veľmi presne ale výherné číslo v lotte je presne opačný prípad. Kvalita predikcie závisí od viacerých faktorov:

1. Ako dobre rozumieme okolnostiam ovplyvňujúcim predikciu?
2. Koľko dát máme k dispozícií?
3. Ako veľmi podobná bude budúcnosť minulosti?
4. Môže predpoveď ovplyvniť to, čo sa snažíme predikovať?

V prípade príkladu so spotrebou elektrickej energie vie byť predikcia veľmi presná, keďže veľmi dobre poznáme všetky štyri faktory:

1. Spotreba elektrickej energie je veľmi ovplyvňovaná teplotou, menej už obdobiami roka (sviatky, dovolenky a pod.) a ekonomickou situáciou.

2. Bežne je k dispozícií veľké množstvo dát o spotrebe elektrickej energie aj o počasí za dlhé časové obdobia (roky až dekády).

3. Pre krátkodobú predikciu (týždne) je možné tvrdiť, že spotreba elektrickej energie bude veľmi podobná posledným týždňom.

4. Pre väčšinu domácností cena elektrickej energie nezávisí od dopytu, preto predpoveď spotreby nemá takmer žiadny vplyv na správanie používateľov.

V prípade predikcie výmenných kurzov mien je splnená iba jedna podmienka. Máme k dispozícií veľké množstvo dát. Na druhú stranu, faktorom ovplyvňujúcim predikciu rozumieme veľmi málo, budúcnosť môže byť veľmi rozdielna od minulosti kvôli finančným a politickým krízam v krajinách a predpoveď kurzov mien môže veľmi výrazne ovplyvniť kurzy samotné.

V prípade kybernetickej bezpečnosti sa budeme snažiť predikovať časové rady vytvorené najmä z počtu bezpečnostných udalostí zachytených rôznymi senzormi v počítačovej sieti. Zo štyroch spomínaných faktorov vieme splniť s rôznymi problémami približne dva až tri:

1. Vo väčšine prípadov nevieme veľmi dobré popísať okolnosti, ktoré sú podstatné pre predikciu. Nikdy nepoznáme presne stav siete, na začiatku nevieme presne identifikovať motív a cieľ útočníka, kedy a akým spôsobom sa bude snažiť napadnúť sieť. Niektoré veci nám však môžu byť nápomocné. Napríklad novo zverejnená kritická zraniteľnosť nejakého systému nám napovedá, že sa zvýši množstvo pokusov o jej využitie a následnú kompromitáciu systému.

2. Oblasť dát je v tomto prípade zaujímavá, keďže vieme získať veľké množstvo dát z rôznych zdrojov. Pri začínajúcej organizácií musíme čakať istú dobu, kým nazbierame dostatočné množstvo dát na to, aby sme mohli začať nad týmito dátami robiť nejaké predikcie. Žiaľ aj po dlhšom čase sa môže stať, že síce máme dostatočné množstvo dát ale otázna je ich kvalita. Viac sa dostupným údajom a dátovým sadám budeme venovať v kapitole 3.1.

3. V prípade podobnosti budúcnosti s minulosťou vieme z krátkodobého hľadiska povedať, že pokusy o útoky prebiehajú neustále, a teda by sa časový rad mal správať podobne, ale môže to byť veľmi nestále pre veľký počet neznámych faktorov. Z dlhodobého hľadiska sa množstvo internetovej prevádzky zvyšuje veľmi rýchlym tempom, a preto môžeme predpokladať len rastúci trend.

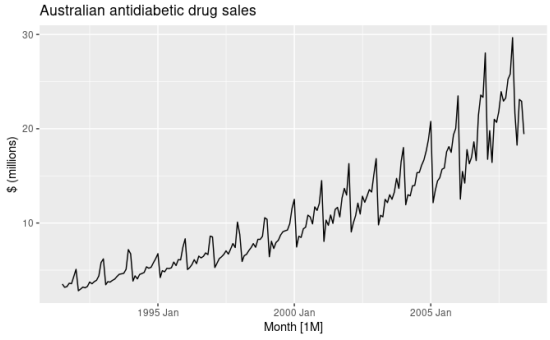
4. V posledom prípade môžeme s istotou tvrdiť, že naše predikcie nijako neovplyvnia správanie útočníkov.

Metódy predikcie veľmi záležia od dát, ktoré sú dostupné. V prípade, že nie sú dostupné dáta alebo dostupné dáta nie sú relevantné pre predikciu, používajú sa kvalitatívne metódy predikcie. My sa však budeme zaoberať kvalitatívnou predikciou, pre ktorú musia byť splnené dve nasledujúce podmienky:

- Musia byť dostupné numerické informácie o minulosti.
- Musí existovať predpoklad, že dôjde k opakovaniu niektorých vzorcov z minulosti alebo ich častí.

Existuje veľké množstvo metód pre kvantitatívnu predikciu, ktoré sú často vytvorené a prispôsobované pre špecifické prípady. Každá metóda má svoje vlastnosti, presnosť, klady a zápory, ktoré je potrebné zvážiť pri výbere správnej metódy. Na rôzne typy metód sa pozrieme v ďalších sekciách.

Pri popisovaní časových radov sa objavujú pojmy ako trend, sezónnosť a cyklickosť, ktoré je potrebné brať do úvahy. Za **trend** sa považuje dlhodobá rastúcosť alebo klesajúcosť v dátach. Nemusí byť len lineárna. V prípade trendu môžeme niekedy hovoriť o „zmene smeru“, kedy dochádza k zmene trendu z rastúceho na klesajúci alebo naopak. Očividný trend v časovom rade je možné vidieť na obrázku 2. O **sezónnosti** môžeme hovoriť, ak je v dátach vidno vplyv sezónnych faktorov, ako sú ročné obdobie, deň v týždni, deň a noc a podobne. Sezónnosť sa vždy spája so známou periódou fixnej dĺžky. Ukážku sezónnosti opäť nájdeme na obrázku 2. **Cyklickosť** je podobná sezónnosti, s tým, že rasty a poklesy v dátach nemajú fixnú frekvenciu.



Obr. 2: Ukážka rastúceho trendu a sezónnosti. Zdroj [37].

Úloha predikcie sa štandardne skladá z piatich základných krokov:

1. **Definícia problému**

   Definícia problému vyžaduje pochopenie toho, na čo bude predikcia použitá, kto predikciu potrebuje a ako bude predikcia použitá organizáciou, ktorá chce predikciu. Analytik, ktorý sa bude zaoberať predikciou sa musí rozprávať s každým, kto bude mať na starosti zber dát, uloženie dát a samotné použitie predikcií.

2. **Získavanie informácií**

   Vždy sú potrebné minimálne dva druhy informácií: štatistické dáta a nahromadené odborné znalosti ľudí, ktorí zberajú dáta a budú používať predikcie. Často je náročné získať dostatočné množstvo historických dát pre tréning dobrého modelu. Niekedy staré dáta nie sú veľmi užitočné kvôli zmenám v systéme, ktorý sa bude predikovať. Vtedy je vhodné použiť len novšie dáta. V každom prípade, dobrý predikčný model by mal byť schopný spracovať aj väčšie zmeny v systéme.

3. **Predbežná analýza**

   Je vhodné začať so zobrazením dát (najčastejšie vo forme grafu). Dôležité je zamerať sa na zodpovedanie nasledujúcich otázok. Je v dátach viditeľný trend, sezónnosť alebo nejaká cyklickosť? Sú v dátach nejaké extrémne hodnoty, ktoré si vyžadujú pozornosť doménového experta? Ako dôležité sú vzájomné vzťahy premenných dostupných pre predikciu?

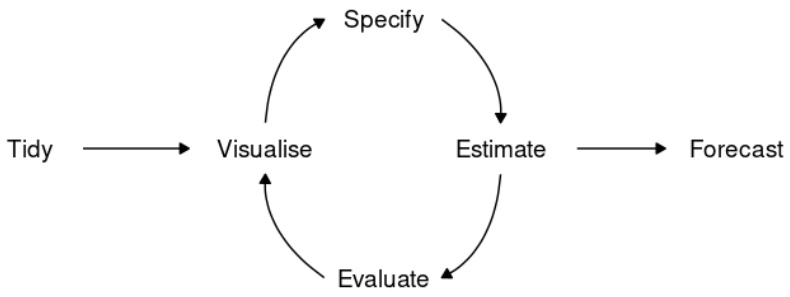
4. **Výber a tréning modelu**

   Výber vhodného modelu závisí od viacerých faktorov: dostupnosť historických dát, dôležitosť vzájomných vzťahov medzi premennými vstupujúcimi do predikcie a vysvetľujúcimi premennými a spôsob, akým bude predikcia použitá. Je bežné vytvoriť a porovnať viacero modelov. Na rôzne typy modelov sa pozrieme v ďalších sekciách.

5. **Predikcia a evaluácia modelu**

   Po výbere modelu, určení jeho parametrov a natrénovaní, sa model použije na samotnú predikciu. Následne je potrebné určiť aký presný je zvolený model. Na rôzne spôsoby vyhodnotenia presnosti predikcií sa pozrieme v ďalších sekciách.

Tento proces sa dá upresniť tak, ako je zobrazené na obrázku 3. Kde Tidy (čistý, upratený) znamená prípravu dát do správneho formátu, ich vyčistenie, identifikovanie chýbajúcich hodnôt, predspracovanie a podobne. Vizualizácia dát (Visualise) je potrebná pre pochopenie dát, identifikovanie vzorcov a výber správneho modelu (Specify). Vybraný model je potrebné natrénovať na vzorke dát (Estimate) a následne zistiť, ako kvalitné predikcie poskytuje (Evaluate). Tento kolobeh sa opakuje, až kým nie je predikcia dostatočná pre naše potreby, resp kým nespĺňa nami vopred zadané požiadavky. Následne už zostáva len používať model na samotnú predikciu (Forecast).



Obr. 3: Popis procesu predikcie od tvorby dát až po samotnú predikciu. Zdroj [37].

2.1 Dáta

V prípade kybernetickej bezpečnosti netvoria časové rady hodnoty, ktoré boli odmerané tak, ako je to napríklad pri počasí (teplota, tlak a pod.). Väčšinou sú časové rady tvorené počtom udalostí, ktoré nastali za určitú dobu. Tieto udalosti môžu byť rôznych typov a z rôznych zdrojov. V práci [91] rozdeľujú dáta v oblasti kybernetickej bezpečnosti do troch vrstiev, ako je možné vidieť na obrázku 4.

Sieťové dáta obsahujú údaje zo sieťovej vrstvy, tak ako ich popisuje TCP/IP model okrem aplikačnej vrstvy TCP/IP modelu. Tento typ dát úzko súvisí so sieťovou aktivitou a možno ich použiť na štúdium rôznych útokov (skenovanie, Dos/DDos a pod.), šírenie škodlivého kódu v sieti a iných sieťových bezpečnostných problémov. Sieťové dáta sa bežne používajú vo forme [91]:

- Sieťových paketov, ktoré sú fyzicky zachytené na sieťovom rozhraní a uložené, keď sieťové rozhraním odovzdáva paket operačnému systému. Na zachytávanie sieťových paketov sa najčastejšie používajú nástroje Wireshark a tcpdump. Štandardne obsahujú časovú pečiatku, IP adresy, porty, veľkosť paketu a podobne.

- Dáta sieťového toku (Network flow data) sú štandardne ukladané vo formáte NetFlow [14]. Je to logovací formát vyvinutý pôvodne firmu Cisco, no je podporovaný na viacerých typoch routrov. NetFlow záznam reprezentuje agregovanú sieťovú komunikáciu medzi dvoma zariadeniami. Záznam obsahuje IP adresy, porty, typ protokolu, množstvo poslaných dát a iné informácie.

- Dáta smerovania sa používajú hlavne na výskum sieťovej topológie. Najčastejšie sa používajú BGP dáta.

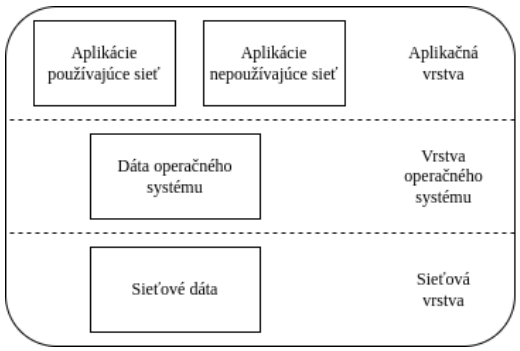
Dáta operačného systému sú najčastejšie informácie, ktoré hovoria o operačnom systéme ako takom. Obsahujú vyťaženosť systémových zdrojov (CPU, pamať a pod.),

informácie o používateľských účtoch a ich oprávneniach, systémove logy a pod. Tieto dáta úzko súvisia s aktivitami v rámci samotného koncového zariadenia.

Najvyššia aplikačná vrstva sa skladá z dvoch tried [91]:

- Dáta generované aplikáciami používajúcimi sieť ako sú napríklad emailové a webové logy.
- Dáta generované aplikáciami, ktoré nepoužívajú sieť ako napríklad databázové logy.

Tieto dáta sú úzko späté aplikáciami a používajú sa na výskum ich bezpečnosti.



Obr. 4: Vrstvy dát v oblasti kybernetickej bezpečnosti. Zdroj [91].

V tejto časti je nutné spomenúť aj konkrétne často používané zdroje dát, ako sú systémy na detekciu prienikov (IDS, Intrusion Detection System) a honeypoty. IDS systémy [97] sú zariadenia alebo programy, ktoré monitorujú sieť alebo nejaký systém na podozrivé aktivity alebo porušenie zadefinovaných pravidiel. Vniknutia alebo porušenia sú reportované administrátorovi alebo sú centrálne ukladané v SIEM systémoch (Security Information and Event Management). SIEM systém kombinuje výstupy z viacerých zdrojov a používajú rôzne filtrovacie systémy na rozlíšenie malicióznej aktivity a falošných hlásení. Honeypot [81] je systém navrhnutý tak, aby vyzeral ako legitímny systém, ktorý by mal byť zaujímavý pre útočníkov. V skutočnosti je to však izolovaný, monitorovaný systém, schopný blokovať a analyzovať pokusy útočníkov o vniknutie do systému. Oba spomínané systémy sú veľmi cennými nástrojmi v oblasti sieťového bezpečnostného situačného povedomia.

Takmer všetky vyššie popísané dáta, ktoré sa dajú zbierať v oblasti kybernetickej bezpečnosti, sa dajú použiť pre vytvorenie časových radov na základe počtu výskytov bezpečnostných udalostí, prípadne na základe sieťovej prevádzky. Tieto časové rady sa následne dajú použiť v oblasti sieťového bezpečnostného situačného povedomia na analýzy a predikcie.

2.2 Metódy

Existuje veľké množstvo metód, ktoré sa dajú použiť na predikciu časových radov. Od úplne jednoduchých ako priemerovania hodnôt, naivnej predikcie (ako predikciu uvedie poslednú známu hodnotu), cez štatistické metódy a metódy klasického strojového učenia až po metódy komplexných neurónových sietí, ktoré stále naberajú na popularite. Ak by chceli rôzne metódy používané na predikciu časových radov popísať podrobne, prekročili by sme rozsah požadovaný na tento typ práce. Z tohto dôvodu si v nasledujúcom texte stručne popíšeme vybrané metódy, niekedy aj s príkladom použitia v literatúre.

2.2.1 Štatistické metódy

ARIMA (Autoregressive integrated moving average) spolu s exponenciálnym vyhladzovaním patrí k najpoužívanejším modelom na predikciu časových radov [37]. ARIMA sa zameriava na autokoreláciu v dátach. Vzniká pridaním diferencie do modelu ARMA (kombinácia modelov AR - autokorelácia a MA - klzavý priemer). Dá sa vyjadriť nasledovne:

$$y_t^{(d)} = c + \sum_{i=1}^{p} \phi_i y_{ti}^{(d)} + \sum_{j=1}^{q} \theta_j \varepsilon_{tj} + \varepsilon_t,$$

kde y_t je d krát diferencovaný rad, c je konštanta, p a q sú stupene autoregresie a kĺzavých priemerov, ϕ_i a θ_j sú koeficienty autoregresie a kĺzavých priemerov a sú chyby.

Exponenciálne vyhladzovanie sa zameriava na popis trendu a sezónnosti v dátach[37]. Predikcie vytvorené týmto modelom sú váženými priemermi minulých hodnôt s tým že váhy exponenciálne klesajú so staršími dátami. Najjednoduchší ETS model sa dá popísať nasledovne:

$$\begin{aligned} s_0 &= x_0; \\ s_t &= \alpha x_t + (1-\alpha) s_{t-1}, \ t > 0, \end{aligned}$$

kde α je faktor vyhladenia, $0 < \alpha < 1$.

Model **prophet** bol uvedený spoločnosťou Facebook v roku 2018, pôvodne pre predikciu denných dát s týždennou a ročnou sezónnosťou a vplyvom prázdninových období [37]. Neskôr bol rozšírený pre podporu viacerých typov dát so sezónnosťou. Najlepšie funguje s dátami z dlhého časového obdobia, v ktorých je silný vplyv sezónnosti. Prophet je možné popísať ako nelineárny regresný model takto:

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t,$$

kde $g(t)$ popisuje po častiach lineárny trend, $s(t)$ sezónnosti s rôznymi vzormi, $h(t)$ vplyv prázdninových období a ε_t je biely šum.

TBATS (Trigonometric Exponential Smoothing State) je metóda exponenciálneho vyhladzovania s Box-Cox transformáciami, ARMA modelom pre reziduá a Trigonometrickou sezónnosťou. Slabou stránka modelu je, že v prípade veľkého množstva dát je výpočtovo náročný, ale k jeho silným stránkam patria: možnosť implementácie viacerých sezónností s využitím malého množstva parametrov, vie pracovať aj s neceločíselnou sezónnosťou a vie pracovať s vysokofrekvenčnými dátami. V článku [40] porovnávali modely TBATS, neurónových sietí a ARIMA pri predikcií ceny elektrickej energie.

Existuje mnoho ďalších modelov, ktoré by sa dali rozoberať, no často sú to len rozšírenia iných modelov ako napríklad SARIMA(Seasonal Autoregressive Integrated Moving Average), SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors) [89]. Za zmienku stoja napríklad aj Grey models [6], vektorová autoregresia alebo bootstrap [37].

2.2.2 Metódy klasického strojového učenia

LightGBM (Light gradient boosting machine) je gradient boosting framework, ktorý používa na stromoch založené učiace algoritmy. Bol predstavený firmou Microsoft v roku 2017 [41] a kvôli svojej rýchlosti a výkonu je používaný pre úlohy regresie, klasifikácie a podobne. V článku [108] ukazujú, že LightGBM je rýchlejší a presnejší ako porovnávané metódy.

XGBoost (Extreme Gradient Boosting) je ďalšia varianta gradient boosting frameworku, ktorá sa používa na rovnaké účely. XGBoost je o niečo pomalší ale dosahuje podobné výsledky. Použitý bol napríklad v práci [31] na predikciu predajov.

SVR (Support Vector Regression) používa rovnaké princípy ako SVM (Support Vector Machine) pre klasifikáciu len s niekoľkými malými rozdielmi. V článku [20] používajú variácie SVM na predikciu rýchlosti vetra.

2.2.3 Metódy neurónových sietí

V prípade neurónových sietí nebudeme popisovať základné architektúry sietí ako perceptrón, plne prepojené dopredné siete, konvolučné alebo rekurentné siete ale zameriame sa skôr na existujúce typy (aj zložitejších sietí) a pridáme odkazy na literatúru, v ktorej boli použité. Niektoré prístupy používajú v prípade len jednoduché málovrstvové **dopredné siete** [109, 52] a ich variácie ako wavelet neural network [106, 32, 45] alebo používajú radialne bázicke funkcie [107]. Iné práce používajú jednu alebo niekoľko vrstiev **rekurentných sietí** [27], respektíve variácie **LSTM** (Long Short-Term Memory) [77, 25] alebo **GRU** (Gated Recurrent Unit) sietí [27]. S pokrokom v oblasti

spracovania prirodzeného jazyka a konvolučných sietí sa začali aj v oblasti predikcie časových radov používať siete pôvodne navrhnuté pre iné oblasti. Príklady takychto sietí sú napríklad siete typu **enkóder-dekóder** [22], **transformer** [101], **informer** [110], **WaveNet** [7] a podobne. V posledných rokoch vznikajú nové typy sietí, ktoré sú priamo určené pre úlohu predikcie časových radov ako sú **N-Beats**, **DeepAR** [50], **Temporal Fusion Transformer** [99, 51].

Metód predikcie časových radov je oveľa viac ako sme teraz popísali. Ďalšie práce, ktoré sa venujú predikciu časových radov v oblasti informačnej a kybernetickej bezpečnosti sú popísané v tabuľke 3.

3 Tézy dizertačnej práce

Ako sme popísali v kapitole 1, sieťové bezpečnostné situačné povedomie je jednou z troch oblastí predikcie v kybernetickej bezpečnosti. Množstvo článkov venujúcich sa tejto oblasti ukazuje potrebu ďalšieho výskumu a experimentovania. Identifikovali sme dvojicu problémov, ktorým sa v tejto oblasti bude potrebné venovať. Jedným je dátová sada a druhým sú metódy predikcie, ktoré sa používajú v tejto oblasti. Kvalitná a dostatočne veľká dátová sada je dôležitá pri každej úlohe strojového učenia. Tabuľka 2 a nasledujúca sekcia však ukazujú, že v tejto oblasti nie je dostupná kvalitná a voľne dostupná dátová sada. Tabuľka 3 a druhá sekcia v tejto časti zas ukazujú, že existuje ešte veľa možností pre zlepšenie predikcií v oblasti sieťového bezpečnostného situačného povedomia, pretože existujúce výskumy pracujú len s jednoduchými a malými dátovými sadami a používajú len jednoduché metódy predikcie časových radov. Hlavný výskumný cieľ tejto práce by sa teda dal sformulovať nasledovne: *Zlepšenie predikcie v oblasti sieťového bezpečnostného situačného povedomia pomocou vytvorenia a zverejnenia kvalitnej dátovej sady, použitia externých údajov a augumentácie dát pri samotnej predikcii a skladania modelov a použitia neštandardných stratových funkcie pri tréningu neurónových sietí.*

3.1 Dátová sada

V dnešnej „dobe neurónových sietí“ je potrebné mať pre akúkoľvek úlohu spojenú so strojovým učením dostatočne veľkú a kvalitnú dátovú sadu. V oblasti počítačovej bezpečnosti vnímame tento problém a preto môžeme jeden z cieľov tejto práce zhrnúť nasledovne: *Vytvorenie testovacej (benchmark) dátovej sady pre oblasť sieťového bezpečnostného situačného povedomia.* Dôvod, ktorý nás doviedol k tomuto cieľu, je takýto. Existujúca literatúra nám ukazuje mnoho autorov, ktorí používajú rôzne dátové sady vo svojich prácach, v ktorých predstavujú rôzne metódy predikcie časových radov s rôznou úspešnosťou. Žiaľ, mnoho z týchto dátových sád má rôzne problémy, ktoré si popíšeme v nasledujúcej podkapitole. Okrem toho, mnoho týchto riešení, je nemožné zreprodukovať a teda naozaj určiť, ich kvality, keďže autori nechcú alebo nemôžu zverejniť dáta s

ktorými pracovali, či už je to z právnych alebo iných dôvodov.

3.1.1 Aktuálny stav riešenej problematiky

V oblasti predikcie časových radov v oblasti kybernetickej bezpečnosti sa stretávame s troma hlavnými typmi časových radov:

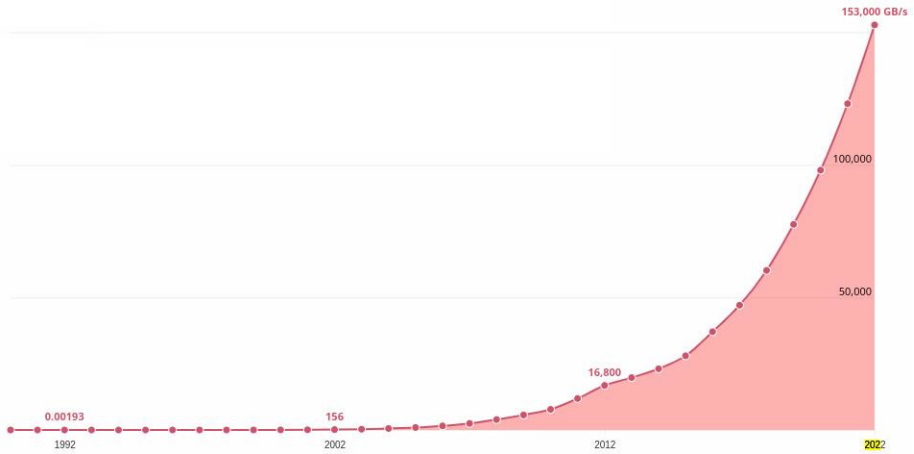
- časové rady vytvorené z reálnej prevádzky siete,
- časové rady vytvorené z dát zozbieraných z honeypotov a podobných senzorov,
- časové rady vytvorené z databáz zraniteľností.

V našej práci sa budeme venovať najmä druhému typu.

V oblasti sieťového bezpečnostného situačného povedomia máme dnes k dispozícií viacero dátových sád. Ich prehľad môžeme nájsť v tabuľke 2. Žiaľ obsahujú viacero problémov:

- vek,
- veľkosť,
- nedostupnosť,
- ...

Množstvo dát, ktoré „pretečú“ internetom sa podľa viacerých zdrojov[15, 96, 98] (dobry zdroj o internetovej prevadzke?) každý rok znásobuje. Graf rastúcej internetovej prevádzky je na obrázku 5. Veľa dátových sád, ktoré sa používajú v oblasti kybernetickej bezpečnosti, bolo vytvorených pred rokom 2015. Podľa nás nie je často možné takéto dátové sady dnes považovať za vhodné pre výskumné účely, lebo charakter internetovej prevádzky sa za 7 rokov už veľmi zmenil.



Obr. 5: Graf rýchlo rastúcej internetovej prevádzky. Zdroj[98].

Ďalším problémom, ktorým trpia dátové sady je ich dĺžka. Z tabuľky 2 je možné vidieť, že väčšina dátových sád obsahuje záznamy maximálne v jednotkách týždňov. Ako sa ukazuje, pre neurónové siete je veľmi dôležitý dostatok dát a dá sa povedať, že čím viac dát, tým lepšie výsledky neurónové siete dosahujú.

Tretím hlavným problémom je nedostupnosť dátovej sady verejne. To znemožňuje vykonanie ďalších výskumov nad popisovanou dátovou sadou, overenia výsledkov dosiahnutých vo vedeckej práci, v ktorej používajú nezverejnenú dátovú sadu a podobne. Motívov prečo autori nezverejňujú dáta s ktorými pracujú, alebo ich zverejňujú iba čiastočne, je viacero. Napríklad dáta pochádzajú z reálnej prevádzky, ktorú nie je možné zverejniť kvôli bezpečnosti, porušovali by tým súkromie a podobne.

Okrem toho dátové sady obsahujú ďalšie problémy ako napríklad, nedostatočný popis dát, nedostatočný popis infraštruktúry na na ktorej boli dáta získavané a podobne.

V [53] popisujú vlastnosti, ktoré by mala mať ideálna dátová sada určená pre validáciu IDS (Intrusion Detection System), takto:

- **príznaky** - Dátové sady by mali obsahovať príznaky zo sieťovej prevádzky (časové pečiatky, počet bitov a paketov, ip adresy, porty a pod.) ale aj z klientov (počet neúspešných pokusov o prihlásenie, logy a pod.). Najlepšie je ak sú dostupné RAW dáta, v takom prípade môžu použiť výskumníci ľubovoľné príznaky.

- **reálna prevádzka** - Dátová sada má obsahovať reálnu prevádzku, minimálne prevádzku v pozadí. Syntetické generovanie môže viesť k nekorektnosti predikčných modelov.

- **prevádzka s reálnymi útokmi** - Dátová sada by mala obsahovať reálne útoky urobené pomocou state-of-art techník a nástrojov.

- **anotovanosť** - Dátová sada by mal mať korektné anotácie prevádzky ako čistej a malicióznej. V prípade útokov by mali byť rôzne anotácie pre rôzne typy útokov.

- **trvanie** - Čas zberania dát by mal byť dostatočne dlhý (dni, týždne, mesiace) aby obsiahol cyklickosť sieťovej prevádzky (deň a noc, pracovný týždeň a víkend).

- **dokumentácia** - Dátová sada musí byť dostatočne zdokumentovaná.

- **formát** - Dátové sady sú štandardne vo formátoch pcap(tcpdump), csv alebo flow(NetFlow). Pcap formát dovoľuje najväčšiu následnú analýzu a používa sa väčšinou pri simulovanej prevádzke, ale nevýhodou je veľkosť dátovej sady, preto sú vhodnejšie pre kratšie časové úseky. Csv súbory vznikajú väčšinou spracovaním

pcap, vybratím zvolených príznakov. Flow dáta sú väčšinou získavané z reálnej prevádzky a sú vhodné aj na zber dát z dlhšieho časového obdobia.

V práci [71] popisujú tieto vlastnosti nasledovne:

- **dynamické generovanie** - Najznámejšie dátové sady ako DARPA98 a DARPA 99 boli vytvorené už pred viac ako dvadsiatimi rokmi. Každá statická dátová sada prestane byť v istú dobu aktuálny a teda už nebude odzrkadľovať aktuálny stav internetovej prevádzky. Preto je potrebné generovať nové dátové sady, ktoré aktuálny stav siete a správania používateľov.

- **reálne dáta** - Dátová sada by mal obsahovať skôr reálnu prevádzku ako generovanú prevádzku. Veľké množstvo faktorov ako čas odozvy servera, úzke hrdlá pripojenia a iné šumy sa generujú veľmi ťažko. Taktiež reálna prevádzka môže obsahovať neznáme útoky, čo je tiež žiadúce pre obsiahlu analýzu.

- **topológia siete** - Pri nasadzovaní IDS je topológia dôležitá. Počítačová sieť v malých alebo stredných spoločnostiach sa zásadne líši od veľkých sietí. Taktiež rozloženie operačných systémov je veľmi dôležité. Niektoré kancelárske siete obsahujú len Windows zariadenia, iné väčšie siete môžu obsahovať rôzne Linuxové servery a mix zariadení s rôznym operačným systémom (Windows, Linux, Mac OS, Android a pod.), každý s inými zraniteľnosťami a možnosťami útokov. Pre vytvorenie dobrej dátovej sady je potrebné toto všetko zobrať do úvahy.

- **normálne správanie používateľov** - Honeypot dátové sady sa skladajú prevažne zo škodlivej prevádzky. Dobrá dátová sada by mala obsahovať normálnu používateľskú prevádzku lebo väčšina sieťovej prevádzky v spoločnosti je normálna a úlohou IDS je identifikovať nebezpečné aktivity vo veľkom toku dát v sieti.

- **špecifické dáta** - Ďalšou požiadavkou je prítomnosť predpokladaných scenárov útoku. Napríklad, ak má byť testovaný algoritmus na detekciu skenovania portov, dátova sada by mala primárne obsahovať skenovania portov ako nebezpečnú aktivitu. Dobrá dátová sada by mal byť prispôsobená na aktivitu, ktorú má overovať.

- **anotovanosť** - Interpretácia sieťovej prevádzky je pre tretie strany veľmi náročná a preto musí byť dátová sada dobre popísaná a musí obsahovať informácie o zariadeniach pripojených do sieti. Keďže mnoho metód detekcie zneužitia je založených na dolovaní dát, dátové sady, ktoré sa používajú musia byť korektne anotované.

- **verejnosť** - Dátová sada by mala slúžiť pre porovnanie rôznych algoritmov. To môže byť splnené len vtedy, ak dátová sada bude verejne dostupná a rôzni výskumníci budú môcť zisťovať kvalitu dátovej sady a používať ju na testovanie svojich algoritmov.

Popísané vlastnosti sú však definované pre IDS datasety. V nasledujúcej časti definujeme takéto vlastnosti, ktoré by mali spĺňať dátové sady pre oblasť predpovedania sieťového bezpečnostného situačného povedomia pomocou predikcie časových radov.

rok	názov	odkaz	problém	typ dostupných dát	typ dátovej sady	dĺžka
1999	KDD Cup 1999 Data	[88]	vek, dĺžka	csv	generované útoky	-
2001	Know Your Enemy: Statistics	[67]	vek	snort alerts, firewall logs	jeden honeynet	$\leq$ 11 m
2010	ISCXIDS2012	[79]	vek, dĺžka	pcap	generované útoky	$\leq$ 7 d
2011	CTU-13 Dataset	[29]	vek, dĺžka	pcap	generované útoky	$\leq$ 70 h
2011	Hackmageddon	[63]	nie situácia	raw	záznamy z veľkých incidentov	-
2013	DDS Dataset Collection	[75]	vek, dĺžka	csv	AWS honeypoty	$\leq$ 6 m
2014	SSH datasets	[33]	vek, dĺžka	logs, NetFlow	honeypot	2m
2015	4SICS 2015	[58]	vek, dĺžka	pcap	industriálne siete	$\leq$ 3 d
2015	UNSW-NB15 Dataset	[55]	vek, dĺžka	csv, pcap	generované útoky	$\leq$ 32 h
2015	AWID2	[42]	vek, dĺžka	csv, pcap	generované útoky	$\leq$ 4 d
2016	ISOT CID	[61]	dĺžka	csv, pcap	generované útoky	$\leq$ 11 d
2016	UGR'16	[53]	dĺžka	csv, NetFlow	generované útoky	$\leq$ 4 m
2017	2017-SUEE-data-set	[38]	dĺžka	pcap	generované útoky	$\leq$ 8 d
2017	CIC-IDS2017	[78]	dĺžka	csv, pcap	generované útoky	$\leq$ 5 d
2017	MITH?FADOCE	[3]	nie situácia	csv	reálny počet incidentov v sieti	$\leq$ 366 t
2017	CIDDS-001	[72]	dĺžka	csv, logs	generované útoky	$\leq$ 4 t
2017	CIDDS-002	[71]	dĺžka	csv, logs	generované útoky	$\leq$ 2 t
2018	UHANDS	[87]	dĺžka	csv, json	sieťová prevádzka	$\leq$ 90 d
2018	WUSTL-IIOT-2018	[85]	dĺžka	csv	industriálne siete	$\leq$ 25 h
2018	CSE-CIC-IDS2018 on AWS	[60]	dĺžka	csv, pcap, logs	generované útoky	$\leq$ 3 d

2018	CIC-DDoS2019	[59]	dĺžka	csv, pcap	generované útoky	$\leq$ 3 d
2020	The TON_IoT Datasets	[56]	dĺžka	csv, pcap	generované útoky	$\leq$ 30 d
2021	Hornet	[44]	dĺžka	NetFlow, csv	honeypot	$\leq$ 40 d
2021	AWID3	[11]	dĺžka	csv, pcap	generované útoky	$\leq$ 30 d

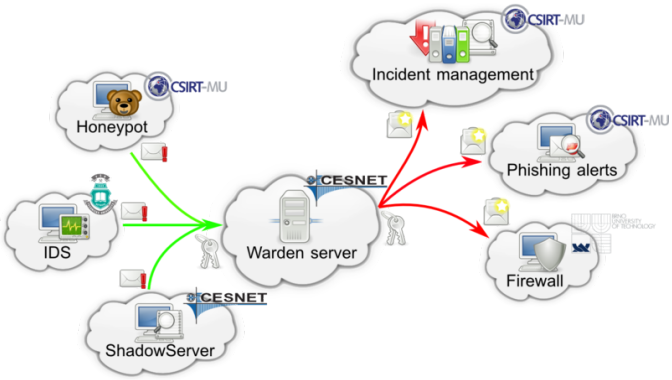
Tabuľka 2: Tabuľka existujúcich dátových sád v oblasti kybernetickej bezpečnosti. Vysvetlenie skratiek: UHANDS - Unified Host and Network Data Set, MITH?FADOCE - Malware in the Future? Forecasting Analyst Detection of Cyber Events, m - mesiacov, d - dní, h - hodín, t - týždňov

3.1.2 Prístup k riešeniu výskumného cieľa

Pri riešení tohoto výskumného cieľa nadviazujeme na prácu [65]. V tejto práci popisujú systém WARDEN [39], generovanie časových radov z neho a aj ich následnú predikciu.

WARDEN je systém pre zdieľanie informácií o detegovaných bezpečnostných udalostiach medzi organizáciami zapojenými do systému WARDEN. Aktuálne je vyvíjaný, testovaný a prevádzkovaný najmä pre potreby siete národného výskumu a vzdelávania CESNET2.

Systém sa skladá z hlavného **Warden serveru**, ktorý zabezpečuje hlavné činnosti a klientov. Klienti môžu byť 2 typov. **Odosielajúci klient** sa stará o odosielanie informácií od zapojenej organizácie na Warden server. **Prijímajúci klient** sa stará o získavanie informácií, ktoré požaduje zapojená organizácia. Serverová strana systému Warden sa stará o prijímanie a ukladanie informácií posielaných od klientov a umožňuje aj prístup ku všetkým platným uloženým udalostiam. Jednoduchý nákres architektúry systému WARDEN je na obrázku 6



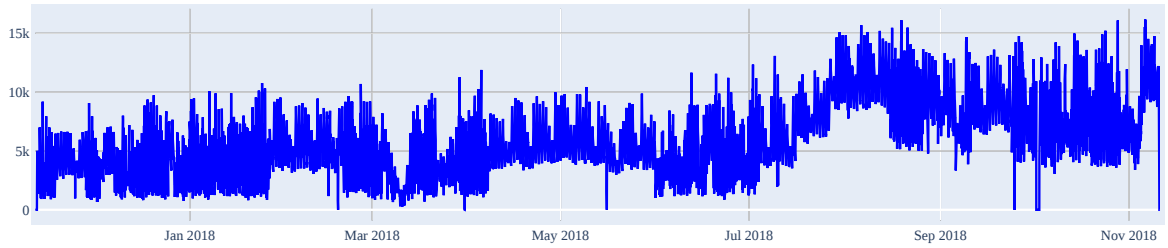
Obr. 6: Architektúra systému WARDEN [92].

Udalosť v systéme WARDEN predstavuje informáciu o zdroji detegovanej bezpečnostnej udalosti niektorou zo zapojených organizácií. Informácie sú získavané rôznymi spôsobmi. Napríklad z detekčných systémov prevádzkovaných v zapojených organizáciách (IDS, honeypoty, monitorovanie útokov na SSH a pod.) alebo z dát z tretích strán (Shadowserver, Honeynet a pod.) alebo z už agregovaných/korelovaných dát. Udalosti sú ukladané vo formáte IDEA. IDEA formát je v podstate json súbor, so štyroma povinnými atribútmi (formát, ID, čas detekcie, kategória). Ďalšími zaujímavými atribútmi sú: kategória, sieťová identifikácia zdroja a cieľa (IP adresa, port, protokol), čas detekcie, čas výskytu.

Spomínaná práca mala k dispozícií dáta zo systému WARDEN z obdobia 11.12. 2017 až 11.12.2018. Na základe rôznych kritérií bolo vytvorených 21 časových radov:

- Count of all alerts,
- Count of unique IP,
- Category recon scanning,
- Category availability DDoS,
- Category attempt login,
- Category attempt exploit,
- Category malware ransomware,
- Category intrusion botnet,
- Port 21,
- Port 22,
- Port 23,
- Port 25,
- Port 80,
- Port 443,
- Port 445,
- Protocol TCP,
- Protocol SSH,
- Protocol UDP,
- Protocol ICMP,
- Protocol Microsoft WBT Server,
- Protocol telnet.

Kritéria boli vyberané zo všetkých možných atribútov kategórie, portu a protokolu. Kritérium muselo byť zastúpené najmenej v 1% údajov. Následne bolo odstránených posledných 6 časových radov z dôvodu, ze obsahovali príliš malé hodnoty, resp. boli nulové z veľkej časti. Tieto časové rady boli použité vo výskume [64]. Testovali dve rôzne dĺžky dátovej sady, jeden a dva mesiace (celá dátová sada mala jeden rok) a ako časovú jednotku v časovom rade používali 30 a 60 minút. Tiež sme ich použili v našom výskume, ktorý je popísaný v nasledujúcej kapitole. Ako časovú jednotku pre časové rady sme už používali 30 minút a používali sme celú dátovú sadu. Takto vytvorené časové rady obsahovali 17 473 hodnôt. Ako je možné vidieť na obrázku 7, takto vytvorené časové rady mali drobné problémy ako napríklad chýbajúce resp. nulové hodnoty na začiatku a na konci a chýbajúce hodnoty v októbri. Okrem toho majú tieto časové rady ešte problém v tom, že pri ich tvorbe sa nepozeralo na množstvo senzorov, ktoré posielali dáta do systému WARDEN. Pridávanie alebo odoberaní odosielajúcich klientov v systéme WARDEN môže spôsobiť veľké zmeny v časovom rade a veľmi nepriaznivo ovplyvniť presnosť predikcie.



Obr. 7: Ukážka časového radu vytvoreného podľa kritéria port 445/TCP(SMB) s časovou jednotkou 30 minút.

V rámci splnenia tohoto cieľa by sme chceli vytvoriť tri nové dátové sady pre predikciu časových radov a zverejniť ich ako testovacie (benchmark) dátové sady.

Prvá dátová sada by bola vytvorená z dát, popísaných vyššie. Časové rady by sme chceli vygenerovať znovu s ohľadom na to, aby bol počet odosielajúcich klientov stále rovnaký.

Aktuálne už máme z WARDENu dostupné takmer trojročné dáta. Z týchto dát by sme chceli vytvoriť nové časové rady podobne ako predchádzajúce. Tie by tvorili druhú dátovú sadu.

Okrem toho, na univerzite prevádzkujeme telekom T-Pot[86]. Je to honeynet platforma, ktorá funguje na viacerých architektúrach, ktorá podporuje viac ako 20 honeypotov, mnoho možností vizualizácie použitím technolgie Elastic Stack, animované mapy útokov a obsahuje viacero bezpečnostných nástrojov.

Z takto zozbieraných dát by sme tiež chceli vytvoriť tretiu dátovú sadu. Takáto dátová sada by mala ďalšie pridané výhody pre našich lokálnych administrátorov a bezpečnostných analytikov, keďže by na základe nej mohli upravovať vlastnosti siete. Okrem toho, by predikcie časových radov vytvorených z dát z T-Potov mohli pomôcť lepšie chrániť lokálnu sieť pred útokmi.

Takto vytvorené dátové sady by sme chceli zverejniť ako voľne dostupné testovacie (benchmark) dátové sady podobne, ako sú vytvorené dátové sady pre iné oblasti strojového učenia ako napríklad ImageNet[19] pre oblasť klasifikácie obrázkov. Dobá dátová sada by však mala mať isté vlastnosti. V predchádzajúcej podsekcií sme uviedli vlastnosti, ktoré by mali spĺňať IDS dátové sady podľa výskumov [53] a [71]. Na základe týchto vlastností sme odvodili vlastnosti, ktoré by mala spĺňať dátová sada časových radov pre oblasť sieťového bezpečnostného situačného povedomia:

- **topológia siete** - Síce topológia siete, z ktorej boli získavané pôvodné dáta, z ktorých boli vytvorené časové rady, nemusí byť dôležitá pre samotnú predikciu, ale je veľmi dôležitá pre to, aby človek, ktorý bude pracovať s dátovou sadou a vytvárať predikčné modely mal prehľad o tom, aký bol stav siete v čase zbierania dát, aké zariadenia a systémy boli použité na zber dát a podobne a na základe toho vedel správne zvoliť predikčné modely a ich parametre.

- **dynamické generovanie, aktuálnosť dátovej sady** - Ako vidieť v tabuľke 2 mnoho dátových sád je staršieho dáta a nereflektujú aktuálny stav sieťovej prevádzky a ani bezpečnostných hrozieb. Aktuálnosť dátovej sady je teda nutnosť a dá sa povedať, že čím novšia, tým lepšia. Okrem toho, dátovú sadu je potrebné neustále aktualizovať novými dátami a pomocou nich zlepšovať aj predikčné modely.

- **veľkosť** - Veľkosť dátovej sady je jeden z najdôležitejších parametrov. Hlavne pre oblasť neurónových sietí platí, že čím viac dát, tým lepšie výsledky model dáva. Veľkosťou dátovej sady sa budeme viac zaoberať v nasledujúcom texte.

- **anotácie** - V prípade dátovej sady určenej na predikciu časových radov nie je nutná presná anotácia jednotlivých pozorovaní. V každom prípade je nutné mať anotované špeciálne udalosti, ktoré nastali v infraštruktúre, z ktorej je dátová sada nakoniec vytváraná. Napríklad, ak dôjde k výpadku niektorého zariadenia na istú dobu, mala by byt táto časť časového radu anotovaná, lebo v nej môže dôjsť k poklesom, ktoré by bez tejto anotácie mohli byť mätúce.

- **verejnosť** - Ako je vidieť v tabuľke 3, viacero prác používa na výskum vlastnú dátovú sadu. Často je táto dátová sada popísaná z pohľadu nastavenia infraštruk-

túry, na ktorej vznikla, no jej nezverejnením dochádza k nemožnosti presného zopakovania daného výskumu a overenia jeho výsledkov. Zverejnenie dátovej sady teda pokladáme za veľmi dôležitú z pohľadu výskumu. Znemožňuje overenie výsledkov existujúcich výskumov ale aj ďalší výskum na danej dátovej sade, a teda celkový pokrok v danej oblasti. Otázna je ešte forma zverejnenia dátovej sady. Zverejneniu RAW dát nedávame vysokú dôležitosť. Či už kvôli častej nemožnosti zverejnenia kvôli ich veľkosti (niekoľkoročné dáta môžu pohybovať v terabajtoch), náročnosti na spracovanie alebo nejednoznačnosti následného spracovania. Za dôležitejšie pokladáme zverejnenie časových radov s čo najmenšou časovou jednotkou (väčšie časové jednotky sú ľahko vytvoriteľné z menších) a presný popis vytvorenia časových radov.

Dnešné hlboké neurónové siete dosahujú svoje výsledky kvôli kvalite, ale najmä kvôli veľkosti dnešných dátových sád. Problém veľkosti dátových sád v oblasti predikcie časových radov rozoberajú v práci [5]. Opisujú 43 prác, ktoré sa zaoberajú predikciou časových radov v rôznych oblastiach (elektrická energia, doprava, dopyt a pod.). Súčasne nastoľujú otázku, aké množstvo dát je potrebné pre spomenuté oblasti, ak sa použijú neurónové siete pre predikciu. K tejto otázke je potrebné prebrať niekoľko dôležitých bodov.

- Množstvo dát v tejto oblasti je často zamieňané s počtom (rozmerom) časových radov. V skutočnosti množstvo dát znamená znamená počet pozorovaní (koľko záznamov má časový rad na osi x). Napríklad zo senzoru, ktorý neustále sleduje nejakú hodnotu meniacu sa v čase, vieme pri sekundovom zaznamenávaní počas jedného roka získať časový rad, pomocou ktorého dokážeme natrénovať komplexnú neurónovú sieť.

- Pravdepodobne je lepšie pozerať na množstvo dát z hľadiska informačného množstva. Napríklad vo finančníctve množstvo informácií v mnoho miliónoch transakcií za hodinu je veľmi limitované množstvom šumu. Naopak v prípade obchodu, kde predaje sledujú sezónnosť a vzory, je možné skôr použiť neurónové siete.

Podľa práce [5] je množstvo dát najlepšie popisovať ako počet pozorovaní, ale niektoré oblasti môžu aj tak obsahovať malé množstvo informácií v týchto pozorovaniach a je náročné použiť neurónové siete v porovnaní s robustnými lineárnymi modelmi. Z praktického hľadiska sa ukazuje, že neurónové siete dosahujú dobré výsledky v oblasti predpovedania dopytu, ak časové rady obsahujú aspoň 50 000 pozorovaní a v oblasti predpovedania vyťaženia elektrickej siete aspoň niekoľko stoviek pozorovaní. V tejto oblasti je ale potrebný ešte ďalší výskum.

Keď predpokladáme jedno rozmernosť časových radov vytvorených podľa kritérií popísaných vyššie v spojení s problémom veľkosti časových radov z práce [5] objavuje sa takýto výskumný podcieľ: *Akú časovú jednotku zvoliť pre časové rady v oblasti kybernetickej bezpečnosti?*

Táto otázka je dôležitá, lebo ak zvolíme príliš malú hodnotu, tak síce budeme mať v časovom rade veľké množstvo pozorovaní, ale za danú časovú jednotku sa podľa niektorých kritérií môže vyskytnúť veľmi málo udalostí alebo sa nemusia vyskytnúť žiadne udalosti. To môže spôsobiť, že časový rad bude tvorený z veľkej časti malými až nulovými hodnotami. V takomto časovom rade niet veľmi čo predikovať. Aj kvôli tomuto problému neboli niektoré časové rady vytvorené pomocou kritérií vyššie vôbec použité.

Naopak, ak zvolíme príliš veľkú hodnotu, v ktorej síce budeme mať v jednotlivých pozorovaniach vysoké počty udalosti, no takýto časový rad bude obsahovať malé množstvo pozorovaní a jeho predikcia nemusí mať žiadnu pridanú hodnotu pre administrátora. Napríklad, ak zvolíme ako časovú jednotku jeden týždeň, tak v prípade ročných dát získame približne 52 pozorovaní, čo môže byť nepoužiteľné pre tréning niektorých metód predikcie (neurónových sietí) a predikcia stavu sieťového bezpečnostného situačného povedomia či už jednokroková alebo viackroková nemusí byť užitočná pre administrátora.

3.2 Metódy predikcie časových radov v oblasti sieťového bezpečnostného situačného povedomia

Dôležitou otázkou pre oblasť predikcie v bezpečnostnom situačnom povedomí je to, prečo potrebujeme (dobrú) predikciu sieťového bezpečnostného situačného povedomia. Uveďme príklad situácie administrátora informačného systému (sieťového/serverového), ktorý má k dispozícií informácie zo systému podobného Wardenu (IDS, Honeypot, logy a pod.). V prvotnej situácií administrátor nemá k dispozícií žiadnu predikciu situácie v tomto systéme. Automatizovaný systém môže zo spomínaných informácií vytvárať časové rady, a tak získa prehľad o situačnom povedomí v sieti. Pridanie predikcií týchto časových radov mu však dodá ďalšiu pridanú informáciu o možnom budúcom stave, čo môže byť veľmi prospešné, lebo odhadnúť budúcu situáciu v sieti je veľmi ťažké aj pre skúseného administrátora. Keď už sú k dispozícií predikcie môžu nastať dva hlavné prípady. Prvým je ten, že predikcie ukazovali nejaké hodnoty, no realita ukazuje, že sú niekoľko násobne vyššie. To môže ukazovať na to, že v sieti sa deje niečo nezvyčajné a administrátor by mal na to primerane zareagovať kontrolou stavu ďalších systémov a podobne. V druhom prípade predikčný model ukazuje hodnoty,

ktoré sa veľmi vymykajú priemeru (napríklad víkendová prevádzka nejakej inštitúcie má byť podstatne vyššia ako bežná prevádzka cez týždeň). V tomto prípade musí administrátor opäť prijať primerané technické alebo bezpečnostné opatrenia, napríklad nastavením reštriktívnejšej politiky firewalu a podobne.

V rámci zlepšenia kvality predikcie v oblasti sieťového bezpečnostného situačného povedomia máme dva ciele a niekoľko podcieľov:
Zlepšenie predikcie na základe dát:

- *pomocou použitia externých údajov (databáza zraniteľností) ako ďalšieho vstupu pre predikčný model,*
- *pomocou použitia augumentácie dát.*

Zlepšenie predikcie na základe úpravy metód:

- *pomocou skladania modelov (ensemble models),*
- *pomocou použitia neštandardných stratových funkcie pri tréningu neurónových sietí.*

3.2.1 Aktuálny stav riešenej problematiky

V oblasti kybernetickej bezpečnosti nájdeme veľké množstvo článkov, ktoré sa zaoberajú predpovedaním sieťového bezpečnostného situačného povedomia. V tabuľke 3 je prehľad vybraných článkov, ktoré sa nejakou formou zaoberajú sieťovým bezpečnostným povedomím. Zobrazený je rok, v ktorom článok vyšiel, dátová sada, ktorú v práci používajú, prístup, ktorý zvolili a cieľ, ktorý autori sledovali.

Výskumy uvedené v článkoch [84, 83, 73, 66] pracujú s objavenými existujúcimi zraniteľnosťami v rôznych systémoch a snažia sa predikovať počet zraniteľností, ktoré sa objavia v budúcnosti. Tento prístup síce nie je priamo zameraný na sieťové bezpečnostné situačné povedome, ale uvádzame ho najmä preto, lebo počty objavených zraniteľnosti môžu veľmi súvisieť s budúcou situáciou v sieti a môžu pomôcť pri predikcií situácie v sieti.

Články [62, 16, 54] pracujú so sieťovým tokom ako takým. Sledujú a predikujú dátovú prevádzku u poskytovateľov internetu (ISP), ktorá síce priamo nehovorí o sieťovej bezpečnostnej situácií, no jej analýza a predikcia by mohli pomôcť pri Denial of service (DoS) útokov.

Práce, ktoré sa priamo zaoberajú predikciou sieťového bezpečnostného situačného povedomia, vieme podľa delenia na konci prvej kapitoly, rozdeliť do dvoch kategórií:

- Práce zaoberajúce sa predikciou hodnoty funkcie F. (Práce používajúce hierarchickú metódu na vyhodnotenie sieťovej situácie, popísané na konci prvej kapitoly.) F je funkcia, ktorou v prácach [45, 52, 109, 107, 12, 106, 32, 100, 26, 27, 70, 49] vyhodnocujú sieťovú bezpečnostnú situáciu. Problém tohoto prístupu je ten, že každý autor má funkciu F definovanú inak. Od úplne jednoduchého stavu nejakej premennej v sieti až po zložité vyhodnocovanie stavu bezpečnosti na jednotlivých klientskych zariadeniach, serveroch a pod., ich násobenie váhami podľa ich dôležitosti a následne sčítanie pre určenie celkovej situácie a podobne. Po spočítaní F, sa zaoberajú meniacimi sa hodnotami F v čase a snažia sa predikovať sieťové bezpečnostné situačné povedomie pomocou časového radu funkcie F.

- Práce zaoberajúce sa predikciou počtu útokov. (Práce používajúce metódu odhadu intenzity útoku na vyhodnotenie sieťovej situácie, popísané na konci prvej kapitoly.) Tieto práce [104, 94, 95, 3, 25, 68] využívajú najmä dáta z honeypotov a podobných senzorov. Z dát z týchto senzorov opäť vytvárajú časové rady, ktoré predikujú rôznymi metódami. Tejto oblasti sa chceme najviac venovať aj v našej práci.

Zaujímavý prístup je v práci [80], kde sa snažia predikovať kedy nastane bezpečnostný incident. Používajú na to neštandardný prístup analýzy príspevkov na sociálnej sieti Twitter.

rok	článok	dátová sada	prístup	čo robili
2005	[62]	tok cez Sprint IP backbone	ARIMA	predikcia budúceho sieťového toku
2008	[45]	Know Your Enemy: Statistics	Wavelet Neural Network	predikcia F
2008	[52]	Know Your Enemy: Statistics	Backward Propagation network	predikcia F
2012	[109]	vlastná + KDDCUP 99	Backward Propagation network	predikcia F
2012	[16]	tok zaznamenaný dvoma ISP	Holt-Winters, ARIMA, ANN	predikcia budúceho sieťového toku
2013	[107]	vlastná	Backward Propagation network, Elman network, Radial Basis Function	predikcia F
2013	[12]	Know Your Enemy: Statistics	Small-world Echo State Network	predikcia F
2015	[73]	zranitelnosti webovych prehliadacov	ARIMA, EXP	predikcia počtu zraniteľností
2015	[104]	vlastný honeypot	FARIMA, GARCH, Time Series Theory, Extreme Value Theory	predikcia počtu útokov
2016	[106]	vlastná	Wavelet Neural Network	predikcia F
2016	[83]	National Vulnerability Database	GARCH	analýza počtu zraniteľností
2017	[32]	vlastná	Wavelet Neural Network	predikcia F
2017	[84]	National Vulnerability Database	GARCH, ARIMA	analýza a predikcia počtu zraniteľností
2017	[66]	National Vulnerability Database	ARIMA, ANN, SVM	predikcia počtu zraniteľností

2017	[100]	Know Your Enemy: Statistics, DARPA	MEA-BP NN	predikcia F
2017	[26]	Know Your Enemy: Statistics	Elman Neural Network	predikcia F
2017	[94]	Hackmageddon	ARIMA	predikcia počtu útokov
2018	[80]	twiter a dáta organizácii o útokoch	sledovanie emotikonov na sociálnej sieti v oblasti bezpečnosti	predikcia kedy nastane útok
2018	[27]	Lab2000DARPA	Tanh–RNN, LSTM, GRU	predikcia F
2018	[54]	tok zaznamenaný ISP `https://robjhyndman.com/hyndsight/tsdl/`	ARIMA, RNN, DWT, kombinácia	predikcia budúceho sieťového toku
2018	[95]	vlastná	ARIMA	predikcia počtu útokov
2018	[3]	vlastná (počty útokov z DoD CSSP)	Bayesian State Space Model, AR	predikcia počtu útokov
2019	[25]	vlastný honeypot	biLSTM	predikcia počtu útokov
2019	[68]	vlastná (industrial)	ARIMA, Grey model	predikcia počtu útokov
2020	[70]	Real Cyber Defense Dataset (CSE-CICIDS2018)	SAPSO-Elman NN	predikcia F
2021	[49]	vlastná	CLPSO-RBF	predikcia F

Tabuľka 3: Tabuľka prehľadu existujúcich článkov v oblasti predikcie sieťového bezpečnostného situačného povedomia.

3.2.2 Prístup k riešeniu výskumného cieľa

3.2.2.1 Zlepšenie predikcie na základe dát

Externé údaje, ako napríklad príspevky na sociálnej sieti Twitter [80], môžu zlepšiť kvalitu predikcie. Analýza hrozieb a zraniteľností je súčasťou sieťového bezpečnostného situačného povedomia. V tejto práci by sme sa však chceli zamerať na použitie databáz zraniteľností ako externého zdroja pre predikčné modely, a tým zlepšiť predikciu v oblasti sieťového bezpečnostného situačného povedomia. Mnoho prác, napr. [46, 48, 34, 10], sa zaoberá predikciou a analýzou počtu zraniteľností v rôznych oblastiach informatiky. Na rozdiel od týchto prác nechceme predikovať budúce zraniteľnosti v rôznych systémoch, ale použiť znalosť toho, že sa objavila nejaká zraniteľnosť na to, aby sme upravili naše predikcie. Tento výskumný cieľ sa dá rozdeliť na 2 časti:

1. *Má takáto externá informácia nejaký praktický význam?*
2. *Akým spôsobom dať predikčnému modelu túto dodatkovú externú informáciu?*

Žiaľ, v rámci tohoto výskumného cieľa sa nám nepodarilo nájsť literatúru, ktorá by sa daným problémom zaoberala, preto nasledujúce úvahy sú v podstate hypotézami, ktoré by sme v budúcnosti chceli overiť.

V prípade prvej časti pracujeme s hypotézou: Bezpečnostné udalosti a incidenty závisia od viacerých interných a externých vplyvov na organizáciu. Pre prienik do systému, resp. siete je nutné, aby existovala bezpečnostná zraniteľnosť (ide o chybu samotného systému alebo siete), ktorú útočník využije. Predpokladáme teda, že vývoj počtu zraniteľností je možné použiť pri predikcií sieťového bezpečnostného situačného povedomia a to minimálne dvoma spôsobmi v dvoch situáciach. Obe vysvetlíme na príklade Portu 22 a teda časového radu pokusov o prihlásenie na ssh a zraniteľností týkajúcich sa protokolu ssh.

Prvá situácia nastáva, keď sa objaví nová významná zraniteľnosť v tomto protokole. Na základe tejto zraniteľnosti môžeme v najbližšom období očakávať nárast pokusov o prihlásenie na ssh. To, že náš predikčný model bude disponovať informáciou o tejto významnej zraniteľnosti, by malo spôsobiť to, že by mal predikovať rastúcu krivku.

Druhá je opačným prípadom, teda model nepredikuje nič problematické (žiadne výrazne rastúce krivky), no v realite krivky prudko narastajú. Samozrejme, v tomto prípade by mal administrátor daného systému spozornieť a zaviesť príslušné opatrenia, vzhľadom na povahu predikovaných dát. Okrem toho stojí za zváženie pozrieť sa na to, či sa v blízkej budúcnosti neobjaví nová zraniteľnosť, ktorá by vysvetľovala tento

prudký nárast (zero day zraniteľnosť, ktorá môže existovať v čase predikcie, ale nie je verejne známa).

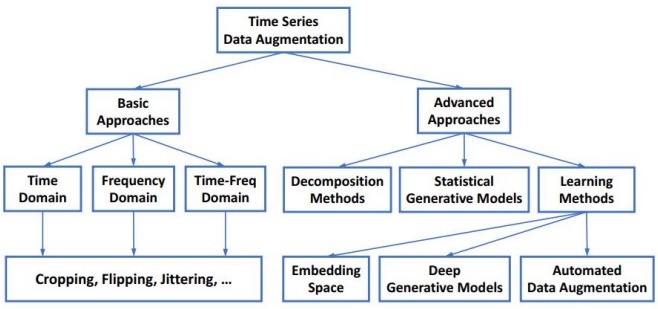
V prípade druhej časti je možné zvoliť viacero prístupov. Najjednoduchším by malo byť vytvorenie viacrozmerného časového radu. Napríklad, zoberme dvojrozmerný časový rad pre Port 22, kde jednou zložkou by boli pokusy o prihlásenie na ssh server a druhou zložkou by boli zraniteľnosti objavené v protokole ssh. Tento prístup je v princípe jednoduchý, použiteľný pre väčšinu existujúcich metód, no jeho problém je v tom, že obe zložky musia mať rovnakú časovú jednotku. Keďže v našom prípade používame najmä 30 minútovú časovú jednotku, tak by najskôr druhá zložka takéhoto časového radu bola vo väčšine prípadov nulová (predpokladáme, že nové zraniteľnosti sa neobjavujú v rádoch hodín). Pre tento prístup by sme teda museli zmeniť veľkosť časovej jednotky na veľkosť aspoň jedného dňa. V prípade, že chceme používať kratšie časové jednotky (v našom prípade napríklad 30 minút), budeme musieť zvoliť iný prístup. Neurónové siete dokážu dostať na vstup dáta rôznych veľkostí. V ich prípade nie je problém dať časti siete 30 minútový časový rad a inej časti siete napríklad počty objavených zraniteľností za posledný týždeň, mesiac a podobne, s rôznou časovou jednotkou. Použitie takéhoto prístupu pre štatistické metódy alebo iné metódy strojového učenia nemusí byť vhodné.

Ďalším problémom v tejto časti je to, aké parametre zvoliť pri vytváraní časového radu zraniteľností. Jednou z databáz zraniteľností, na ktorú sa chceme sústrediť je National Vulnerability Database [9]. Pre hodnotenie zraniteľností používa skóre Common Vulnerability Scoring System (CVSS) [28]. Pre výpočet tohoto skóre je potrebných viacero parametrov, ako napríklad komplexnosť útoku, potrebné oprávnenia, nutnosť používateľského zásahu, aký veľký je dopad danej zraniteľnosti a podobne. To, či bude pri vytváraní časového radu zraniteľností stačiť zobrať len počty nových zraniteľností alebo aj skóre CVSS alebo bude potrebné zobrať aj nejakú kombináciu ďalších parametrov alebo bude nutné zobrať ako vstup pre predikčný model viacero časových radov zraniteľností vytvorených na základe viacerých parametrov bude predmetom ďalšieho štúdia literatúry a empirického skúmania.

Augumentácia dát je v oblasti strojového učenia, najmä neurónových sietí veľmi dôležitá. Používa sa najmä pri nedostatku tréningových dát na vytvorenie nových dát úpravou existujúcich. Napríklad pri práci s obrazovými dátami (klasifikácia, segmentácia a pod. obrázkov) sú na argumentáciu široko používané jednoduché úpravy obrázkov ako zrkadlové otočenie, rotácia, orezanie, skosenie, zmena škály, rôzne transformácie, úpravy farebnosti (prevod do čierno-bielej, úpravy saturácie, svetlosti, expo-

zície a pod.), rozostrenie, pridanie šumu, náhodne výrezy a pod. a ich kombinácie. V oblasti predpovedania sieťového bezpečnostného situačného povedomia pomocou predpovedania časových radov sme sa nestretli s použitím augumentácie aj napriek tomu, že množstvo výskumov pracuje s veľmi malými dátovými sadami. Jedným z dôvodov, že možnosti augumentácie dát nie sú v oblasti časových radov také veľké, jednoduché a priamočiare.

V práci [93] uvádzajú prehľad možností augumentácie časových radov. Na obrázku 8 je uvedená taxonómia metód používaných na augumentáciu časových radov. Popisujú rôzne metódy od najjednoduchších z časovej domény ako pridávanie rôznych šumov, vyrezávanie, škálovanie rôznych oblastí časového radu, zrkadlové pretočenie časového radu a podobne. Pri frekvenčnom spektre popisujú prevod frekvenčného spektra pomocou Furierovej transformácie, kde následne nahrádzajú alebo pridávajú amplitúdové spektrum alebo fázové spektrum rôznymi šumami alebo prezentujú rôzne iné operácie v oblasti Furierovej a inverznej Furierovej transformácie. Z pokročilých metód dekompozície spomínajú napríklad dekompozíciu a úpravu reziduálov a ich následne pridanie do trendu a sezónnosti a tým vytvorenie nových časových radov. Pokročilé prístupy štatistických generatívnych modelov využívajú rôzne metódy ako mix Gaussových stromov, Lokálne a globálne trendy, MAR a pod. na vytváranie nových časových radov. Učiace metódy by mali byť schopné nie len vytvárať podobné vzorky ale byť schopné aj vytvárať nové dáta imitujúce reálne dáta. Nájdeme tu metódy pracujúce s latentným priestorom ako MODALS (Modality-agnostic Automated Data Augmentation in the Latent Space), prístupy podobné GAN modelom, ale aj modely automaticky hľadajúce najlepšie augumentačné prostriedky pomocou reinforcement learning, meta learning alebo evolučných algoritmov. Nie všetky metódy sú však vhodné na augumentáciu časových radov, ktoré majú byť použité na predikciu.



Obr. 8: Taxonómia augumentácií časových radov z práce [93].

Okrem toho evaluujú rôzne metódy augumentácie pre 3 najčastejšie úlohy pri práci s časovými radmi, a to: klasifikácia, detekcia anomálií a predikcia. V prípade predikcie pracujú s piatimi časovými radmi (electricity a traffic z UCI Learning Repository a 3

datasety z M4 súťaže) a dvoma metódami (DeepAR [74] a Transformer vychádzajúci z [90]). Použili základné augumentácie dát (výrezy, zakrivenia a pretočenia) a augumentácie vo frekvenčnej doméne založené na APP. V tabuľke 9 sú zobrazené výsledky, ktoré dosiahli s a bez augumentácií. Ako metriku použili MASE a priemerné relatívne zlepšenie (ARI) počítali ako priemer z $(MASE_{w/o\ aug} - MASE_{w/\ aug})/MASE_{w/\ aug}$. Ako vidieť, nie vždy dosiahli metódy augumentácie lepšie výsledky a táto oblasť si vyžaduje ďalšie skúmanie.

Dataset	DeepAR			Transformer		
	w/o aug	w/ aug	ARI	w/o aug	w/ aug	ARI
electricity	0.87	0.97	1.92%	1.04	1.11	−2%
traffic	0.66	0.80	−12%	0.70	0.91	−16%
m4-hourly	6.33	5.35	56%	7.77	7.87	38%
m4-daily	4.88	4.48	10%	7.85	7.38	37%
m4-weekly	12.00	9.34	76%	6.62	7.09	23%

Obr. 9: Výsledky dosiahnuté augumentáciou časových radov v práci [93].

Existuje množstvo ďalších prác [4, 18], ktoré sa zaoberajú augumentáciou časových radov. V rámci tejto práce by sme chceli vyskúšať existujúce metódy augumentácie časových radov a pomocou nich zlepšiť predpovedanie sieťového bezpečnostného situačného povedomia.

3.2.2.2 Zlepšenie predikcie na základe práce s metódami

Skladanie modelov (ensemble modeling) je proces, pri ktorom je viacero rozdielnych modelov použitých na vytvorenie predikcie. Motivácia, ktorá je skrytá za skladaním modelov, je taká, že tento prístup znižuje chybu generalizácie. Za predpokladu, že modely sú rozdielne a nezávislé, použitím skladania modelov dochádza k znižovaniu chýb predikcie. Zložený model má síce na pozadí viacero modelov, no správa sa ako jeden model. Skladanie modelov sa bežne používa v praktickej dátovej analýze. [43] Žiaľ v oblasti predikcií sieťového bezpečnostného situačného povedomia je použitie skladania modelov zriedkavé. Z tabuľky 3 používajú alebo spomínajú nejakú formu skladania modelov len práce [16, 3].

Používanie skladania modelov je bežné aj v oblasti predikcie časových radov. Existujú rôzne metódy skladania modelov, napríklad rôzne kombinácie predikcií jednotlivých modelov (priemerovanie, vyberanie strednej hodnoty, váhovanie a pod.), kombinovanie rôznych neurónových sietí a ich vlastnosti (architektúr, parametrov, učenie len na častiach trénovacej množiny a pod.), kombinovania rôznych metód pomocou iných metód strojového učenia (boosting metódy) alebo jednoduché použitie predikcie z jedného modelu ako ďalší vstup pre iný model.

V práci [105] predikujú výmenný kurz Britskej libry a Amerického dolára. Používajú rôzne metódy skladania modelov, najmä skladanie modelov s rôzne počiatočne inicializovanými váhami, modelov s rôznymi architektúrami a modelov trénovaných na rôznych častiach dát. Ukazuje sa, že skladanie modelov rôzne počiatočne inicializovanými váhami neprináša výrazne zlepšenie. Na druhú stranu, skladanie modelov s rôznymi architektúrami dosahuje dobré výsledky. Naviac, výsledky ukazujú, že je vhodnejšie skladať modely s trénované na rôzne dlhých vzorkách ako modely s rôznym počtom skrytých vrstiev. Výsledky skladania modelov trénovaných na rôznych častiach datasetu taktiež vyzerajú sľubne.

Starší článok [30] popisujú nový prístup Generalized Regression Neural Network (GRNN) Ensemble for Forecasting Time Series, ktorý spája niekoľko základných GRNN pomocou kombinujúcej GRNN na vytvorenie finálnej predikcie. Ich prístup porovnávajú s jedenástimi inými algoritmami na tridsiatich dátových sadách. GRNN Ensemble for Forecasting Time Series má sice o niečo vyššiu výpočtovú náročnosť, no prekonáva porovnávane prístupy v jedno, päť aj desať krokovej predikcii.

Dvojica burzových indexov a dvojica výmenných kurzov je predikovaná v práci [82]. Dátovú sadu rozdeľujú na časti, na ktorých trénujú viaceré LSTM siete, ktorých výsledky sú následne integrované pomocou AdaBoost algoritmu pre vytvorenie skladaného modelu. Zvolený prístup zlepšuje presnosť predikcie a prekonáva iné jednoduché modely ale aj skladané modely.

Štúdia [17] z roku 2022 porovnáva metódy klasického strojového učenia, hlbokého učenia a skladaných modelov na predikciu počtu slnečných škvŕn. Predstavujú model nazvaný XGBoost-DL, ktorý používa XGBoost ako metódu skladania modelov hlbokého učenia, ktorý dosahuje najlepšie výsledky zo všetkých porovnávaných modelov, dokonca prekonávajú predikcie, ktoré urobila NASA. Ďalšou výhodou tejto práce je, že predstavený model je zverejnený ako python knižnica [103].

Vyššie popísané články uvádzajú zaujímavé výsledky, ktoré ukazujú vhodnosť použitia skladania modelov. Pre splnenie tohoto cieľa by sme chceli vyskúšať ďalšie metódy strojového učenia (napríklad tie, ktoré sú popísané v sekcií 2.2), vyskúšať rôzne popisované metódy skladania modelov a určiť ich potenciál pre predikciu sieťového bezpečnostného situačného povedomia. Tomuto cieľu sme sa už čiastočne venovali v článkoch 1 a 2 (príloha A), kde sme na predikciu použili aj jednoduchú kombináciu (priemer) metód ARIMA a exponencialne vyhladzovanie.

Stratová funkcia určuje, ako veľmi sa výsledok výpočtu neurónovej siete líši od skutočnosti. Väčšina neurónových sietí je trénovaných pomocou stochastického poklesu gradientu. Výber stratovej funkcie je veľmi dôležitý, pretože derivácia stratovej fun-

kcie sa používa pri výpočte gradientu pri tréningu. Stratová funkcia môže ovplyvňovať rýchlosť učenia ale aj výslednú presnosť neurónovej siete. Väčšina prác v oblasti predikcie sieťového bezpečnostného situačného povedomia používa pri tréningu mean absolut error (MAE) alebo mean squared error (MSE). Existujú aj ďalšie stratové funkcie, ktoré sa dajú použiť pri tréningu neurónových sietí, ktoré riešia regresný problém, teda aj predikciu časových radov.

Jeden z cieľov článku [76] je empirickou analýzou určiť kombináciu stratovej funkcie a optimalizátora, ktorá vedie k najpresnejším predikciám veternej elektrickej energie. Z optimalizátorov vyberajú SGD, Adagrad, Adadelta, Adam a Nadam. Zo stratových funkcie MAE, MAPE, MSLE, Hinge loss, Squared Hinge loss, Log-Cosh loss, Binary Cross-Entropy loss, Kullback Leibler Divergence loss, Poisson loss a Cosine Proximity loss. Ich výsledky ukazujú, že ako najvhodnejší optimalizátor je dobré voliť Adadelta [76]. Popisujú, že najčastejšie preferovaná stratová funkcia v literatúre je RMSE. Z ich analýzy však vyplýva, že použitie MAE stratovej funkcie vedie k najpresnejším predikciám. Pri použití inej hodnotiacej metriky vyzerá zaujímavo aj použitie MSE stratovej funkcie.

Práca [13] prináša použitie novej stratovej funkcie pre neurónové siete pre predikciu rýchlosti vetra. Porovnávajú predikciu pomocou obyčajnej rekurentnej siete, LSTM siete a GRU siete s MAE, MSE a ich novou Kernel MSE stratovou funkciou. Kernel MSE stratová funkcia je definovaná pomocou prevodu do RKHS (reproducing kernel Hilbert space), výpočtu vzdialenosti pomocou Mercerovho kernelu a Gausovej kernel funkcie nasledovne:

$$l_{kernel-MSE}(y, \hat{y}) = \sum_{i=1}^{N}(1 - exp(-(y_i - \hat{y}_i)^2/2\sigma^2)) \tag{1}$$

pričom v tejto práci používajú $\sigma = \sqrt{2}/2$. Ukazuje sa, že nová kernel MSE funkcia dosahuje pri tréningu lepšie výsledky a následné predikcie sú presnejšie.

Pre splnenie tohoto cieľa by sme chceli natrénovať rôzne typy neurónových sietí s použitím rôznych stratových funkcii na dátach, ktoré máme k dispozícií a porovnaním výsledkov určiť ich vhodnosť pre predikciu sieťového bezpečnostného situačného povedomia. Stratové funkcie, na ktoré by sme sa chceli zamerať sú napríklad tieto (v zátvorke je uvedené číslo definície príslušnej stratovej funkcie): MAE - Mean Absolute Error (2, MSE - Mean Square Error (3), RMSE - Root Mean Square Error (4), Hubert loss (5), MASE - Mean Absolute Scaled Error (6), RMSSE - Root Mean Squared Scaled Error (7), MSLE - Mean Squared Logarithmic Error (8), Log-Cosh loss (9).

$$l(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2}$$

$$l(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

$$l(y, \hat{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4}$$

$$l_\delta(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(\begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{ak } |y_i - \hat{y}_i| \leq \delta \\ \delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta) & \text{inak} \end{cases}) \tag{5}$$

$$l(y, \hat{y}) = \frac{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|} \tag{6}$$

$$l(y, \hat{y}) = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\frac{1}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1})^2}} \tag{7}$$

$$l(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(log(y_i + 1) - log(\hat{y}_i + 1))^2 \tag{8}$$

$$l(y, \hat{y}) = \sum_{i=1}^{N} log(cosh(\hat{y}_i - y_i)) \tag{9}$$

Kde y sú skutočné hodnoty, $\hat{y}$ sú predikované hodnoty, N je počet vzoriek, y_i je i-ta skutočná hodnota, $\hat{y}_i$ je i-ta predikovaná hodnota, δ je zvolený parameter, T je počet vzoriek v trénovacej množine. Nie všetky popísané funkcie sa bežne používajú ako stratové funkcie a existujú aj mnohé iné stratové funkcie, ktoré bude možné vyskúšať, či už z popísaných článkov alebo z ďalšej literatúry.

Tento cieľ je už čiastočne splnený. V článku 2 (príloha A), ktorý je popísaný v nasledujúcej kapitole sme porovnali MAE, MSE a MASE stratové funkcie pri predikcii 2 časových radov (Port 443/TCP a Count of all allerts). Ukazuje sa, že MAE a MASE stratové funkcie dosahujú podobné výsledky a sú o niečo lepšie ako dosahuje MSE. V článku 1 (príloha A) je popísaná predikcia viacerých časových radov, ktoré mame k dispozícií. Žiaľ, vo väčšine prípadov, vyzerá táto predikcia ako naivná predikcia aj napriek tomu, že boli použité štatistické metódy ako ARIMA a exponenciálne vyhladzovanie alebo komplexné neurónové siete. V rámci tohoto cieľa stojí za zváženie aj vytvorenie stratovej funkcie, ktorá by model penalizovala za predikciu, ktorá je podobná naivnej predikcii. Najbližšie k takejto funkcií má funkcia MASE, ktorej slovný popis je nasledovný: Ak je jej hodnota 1, tak predikcia je približne rovnaká ako naivná predikcia. Ak je jej hodnota vyššia ako 1, tak je horšia ako naivná predikcia a ak je menšia ako 1, tak je lepšia ako naivná predikcia. Ako sme však popísali vyššie, resp v nasledujúcej kapitole, stratové funkcie MAE a MASE majú približne rovnaké výsledky a ostáva teda otázkou, či je možné definovať lepšiu stratovú funkciu, ktorá by model penalizovala za predikciu, ktorá je podobná naivnej predikcii.

4 Parciálne výsledky

V tejto kapitole zhrnieme naše doterajšie výsledky, ktoré sú publikované v prácach 1 a 2 uvedených v prílohe A. V oboch prácach sme nadviazali na článok [64] pracovali s dátovou sadou popísanou v časti 3.1.2.

V článku 1 sme sa venovali najmä porovnaniu výsledkov predikčných modelov na dátovej sade získanej zo systému Warden. Z štatistických metód sme použili naivnú predikciu, ARIMA model, exponenciálne vyhladzovanie a kombináciu ARIMA modelu a exponenciálneho vyhladzovania, každú metódu s a bez metódy posuvného okna. Z modelov neurónových sietí sme zvolili celkom deväť rôznych sietí, konkrétne:

- dopredná sieť so štyrmi skrytými vrstvami (DN),
- rekurentná sieť s troma LSTM vrstvami (LSTM),
- rekurentná sieť s troma GRU vrstvami (GRU),
- konvolučná sieť s troma 1D konvolučnými vrstvami (Conv1D),
- konvolučná sieť s troma 1D konvolučnými vrstvami (iné parametre) (Conv1DS),
- enkóder-dekóder sieť s dvoma a dvoma 1D konvolučnými vrstvami (edConv1D),
- enkóder-dekóder sieť s jednou a jednou LSTM vrstvou (e1d1),
- enkóder-dekóder sieť s jednou a jednou LSTM vrstvou s normalizaciou (e1d1LN),
- enkóder-dekóder sieť s dvoma a dvoma LSTM vrstvami (e2d2).

Pri tréningu neurónových sietí sme použili tri rôzne škálovania dátovej sady: logaritmickú diferenciu inšpirovanú z [57] (log diff), štandardne používané odpočítanie priemeru a podelenie štandardnou odchýlkou (mean std) a neštandardnú metódu zlogaritmovania dát (log). Vplyv škálovania sme porovnali a najlepšie modely neurónových sietí sme porovnali s štatistickými modelmi.

Pre účely tejto práce sme použili 15 vybraných časových radov na základe predchádzajúceho výskumu [64]. Dátovú sadu rozdelili na tri časti. Dĺžka celého časového

	škálovanie	počet
	logdiff	10
1 kroková predikcia	log	3
	meanstd	2
	logdiff	5
2 kroková predikcia	log	3
	meanstd	7
	logdiff	3
5 kroková predikcia	log	4
	meanstd	6
	logdiff	3
10 kroková predikcia	log	5
	meanstd	7

Tabuľka 4: Porovnanie počtov najlepších škalovacích metód.

radu bola 17 473 pozorovaní pri 30 minútovej časovej jednotke. Prvá časť začínala pozícií 28 a končila na 15 549. Druhá bola od 15 603 po 17 159 a tretia od 17 160 do 17 459. Hlavným dôvodom bolo to, že medzi 15 550 a 15 601 boli chýbajúce hodnoty a na začiatku a na konci dátovej sady boli veľmi malé alebo nulové hodnoty. Prvá a druhá časť dátovej sady bola použitá na tréning neurónových sietí. Ako časové okno pre tréning neurónových sietí bolo použitých 144 hodnôt, teda tri dni. Druhá časť bola použitá na tréning štatistických metód s aj bez metódy posuvného okna. Tretia časť bola použitá na validáciu a porovnávanie predikcií jednotlivých modelov.

Zamerali sme sa na jedno, dvoj, päť a desať krokovú predikciu. Pri vyhodnocovaní presnosti jednotlivých metód a ich porovnanie sme použili Diebold-Marino test [21] a metriku MASE (Mean Absolute Scaled Error):

$$\text{MASE} = \text{mean}(|q_j|)$$

$$q_j = \frac{e_j}{\frac{1}{T-1}\sum_{i=2}^{T}|y_i - y_{i-1}|}.$$

Jednoduché vysvetlenie metriky MASE je také, že ak je rovná jedna, tak predikcia je približne rovnaká ako naivná predikcia. Ak je menšia ako jedna, tak je lepšia ako naivná predikcia a ak je väčšia, tak je horšia.

V tabuľke 4 sú počty škálovaní, ktoré dosiahli najlepšie výsledky z modelov neurónových sietí pre každý časový rad. Ako je vidieť, ich rozloženie je približne rovnaké a teda nevieme dôjsť k záveru, ktorá z nich je najvhodnejšia. Log však neodporúčame používať.

V tabuľke 4 je sumár najlepších neurónových sietí pre všetky škálovacie metódy. V každom z riadkov log, log diff a mean std je súčet 15 (15 časových radov). Ako je vidieť priemerne najlepšie výsledky sú z enkóder-dekóder sietí.

		DN	LSTM	GRU	Conv1D	Conv1DS	edConv1D	e1d1	e1d1LN	e2d2
1-step	log diff	1	2	2	0	1	1	3	1	4
	log	0	4	2	2	2	0	1	3	1
	mean std	1	1	3	0	0	2	4	4	0
	sum	2	7	7	2	3	3	8	8	5
2-steps	log diff	0	4	2	0	1	1	4	2	1
	log	0	1	2	4	3	0	2	1	2
	mean std	1	1	1	0	1	2	4	4	1
	sum	1	6	5	4	5	3	10	7	4
5-steps	log diff	0	1	2	0	1	2	4	3	2
	log	0	0	2	2	3	0	5	1	2
	mean std	1	2	1	0	1	3	5	1	1
	sum	1	3	5	2	5	5	14	5	5
10-steps	log diff	0	1	4	1	1	1	2	2	3
	log	0	1	1	1	3	2	3	0	4
	mean std	1	0	0	0	2	4	4	1	3
	sum	1	2	5	2	6	7	9	3	10
in total	log diff	1	8	10	1	4	5	13	8	10
	log	0	5	7	9	11	2	11	5	9
	mean std	4	5	5	0	4	11	17	10	5
	sum	**5**	**18**	**22**	**10**	**19**	**18**	**41**	**23**	**24**

Tabuľka 5: Zhrnutie počtov vybraných modelov neurónových sietí, ktoré dosahovali najlepšiu úspešnosť pre každú škálovaciu metódu.

Porovnania najlepších predikcií štatistických metód a neurónových sietí pre jedno a desať krokovú predikciu sú v tabuľkách 6 a 7. Hrubým písmom je vyznačená najmenšia hodnota MASE. Takmer vo všetkých prípadoch mali neurónové siete lepšie výsledky. Pre prípady dvoj a päť krokovej predikcie boli výsledky veľmi podobné. Diebold-Marino test v tabuľkách 8 a 9 ukazuje, že pre jedno krokovú predikciu boli štatistické metódy horšie v ôsmich prípadoch a pre desať krokovú predikciu v šiestich prípadoch (ak je p-hodnota väčšia ako 0,05 tak predikcie mali rovnakú presnosť, ak menšia, tak predikcia štatistickej metódy bola horšia ako predikcia z neurónovej siete).

kritérium	Count of all alerts	COU IPs	COU IPs (AP)	recon scanning	attempt login
štatistický model	E - 0,6904	E - 0,7336	E - 0,6958	Ew - 0,8058	A - **1,0462**
log diff	Conv1DS - **0,6534**	DN - **0,7054**	GRU - **0,6505**	e1d1 - **0,7690**	GRU - 1,0550
log	e1d1 - 0,6553	Conv1D - 0,7191	Conv1D - 0,6619	e2d2 - 0,7758	LSTM - 1,1027
mean std	LSTM - 0,6605	GRU - 0,7230	e1d1LN - 0,6637	e1d1 - 0,7823	e1d1 - 1,0678

kritérium	attempt exploit	Port 22	Port 23	Port 80	Port 445
štatistický model	Ew - 0,6421	N - 1,6024	AEw - 0,7286	AEw - 1,0964	AE - 0,5920
log diff	e2d2 - 0,6429	edConv1D - **1,5988**	e1d1LN - 0,6964	LSTM - **1,0790**	e2d2 - **0,4209**
log	e1d1LN - **0,6246**	GRU - 1,6950	Conv1DS - 0,6941	LSTM - 1,0944	Conv1DS - 0,4997
mean std	GRU - 0,6260	edConv1D - 1,6476	e1d1LN - **0,6898**	e1d1LN - 1,0974	DN - 0,6418

kritérium	TCP	SSH	ICMP	MS WBT Server	telnet
štatistický model	E - 0,9661	Aw - **1,1321**	Aw - 0,6323	Ew - 1,0141	AE - 0,7789
log diff	e1d1 - **0,8316**	e1d1 - 1,1336	e2d2 - 0,6491	LSTM - 1,0058	e2d2 - 0,7788
log	LSTM - 0,8772	e1d1LN - 1,2428	GRU - **0,6178**	LSTM - 0,9993	e1d1LN - **0,7490**
mean std	e1d1LN - 0,8636	edConv1D - 1,3046	GRU - 0,6179	e1d1 - **0,9847**	e1d1 - 0,7540

Tabuľka 6: Porovnanie najlepších štatistických metód a najlepších neurónových sietí pomocou MASE kritéria pre jedno krokovú predikciu pre všetky časové rady. Označenia: COU - Počet unikátnych; AP - Iný prístup; A - ARIMA model; E - exponenciálne vyhladzovanie; N - naivný model; AE - ARIMA + exponenciálne vyhladzovanie (priemerovanie); w - metóda posuvného okna.

kritérium	Count of all alerts	COU IPs	COU IPs (AP)	recon scanning	attempt login
štatistický model	AEw - 0,9578	Aw - 1,1651	AEw - 0,9492	AEw - 1,0784	A - 3,3308
log diff	GRU - 0,9408	GRU - 1,1880	GRU - 0,8771	GRU - 1,0629	e2d2 - 3,4062
log	e1d1 - 0,8854	e1d1 - 1,0903	Conv1D - 0,8578	GRU - **0,9483**	Conv1DS - **3,3129**
mean std	e1d1 - **0,8169**	e1d1 - **1,0371**	e1d1 - **0,8358**	e2d2 - 0,9760	edConv1D - 3,8772
kritérium	attempt exploit	Port 22	Port 23	Port 80	Port 445
štatistický model	Ew - 0,6453	N - 3,3959	AEw - 1,1520	E - 1,8115	Ew - 0,6520
log diff	e1d1 - 0,6507	edConv1D - **3,3749**	e2d2 - 1,2185	Conv1DS - 1,7963	LSTM - **0,6025**
log	e2d2 - 0,6242	Conv1DS - 3,3820	e1d1 - **1,0883**	e1d1 - 1,8320	Conv1DS - 0,6065
mean std	e1d1 - **0,6210**	edConv1D - 3,7218	e2d2 - 1,0983	Conv1DS - **1,7567**	DN - 0,7080
kritérium	TCP	SSH	ICMP	MS WBT Server	telnet
štatistický model	A - 1,1029	Aw - **3,6500**	Aw - 0,6300	AE - 1,8209	AE - 0,8734
log diff	e1d1LN - 1,0453	Conv1D - 3,6610	e2d2 - 0,6770	e1d1LN - 1,8096	e1d1 - 0,9549
log	e1d1 - **1,0144**	Conv1DS - 3,9079	e2d2 - 0,6278	e2d2 - 1,8269	e2d2 - **0,8460**
mean std	e1d1LN - 1,0584	edConv1D - 4,1247	e2d2 - **0,6267**	Conv1DS - **1,7559**	edConv1D - 0,8750

Tabuľka 7: Porovnanie najlepších štatistických metód a najlepších neurónových sietí pomocou MASE kritéria pre desať krokovú predikciu pre všetky časové rady. Označenia: COU - Počet unikátnych; AP - Iný prístup; A - ARIMA model; E - exponenciálne vyhladzovanie; N - naivný model; AE - ARIMA + exponenciálne vyhladzovanie (priemerovanie); w - metóda posuvného okna.

Time series	Stat	Neu	p-value
Count of all alerts	E	log diff Conv1DS	0,0200
COU IPs	E	log diff DN	0,1914
COU IPs (AP)	E	log diff GRU	8,5e-05
recon scanning	Ew	log diff e1d1	0,0110
attempt login	A	log diff GRU	0,6613
attempt exploit	Ew	log e1d1LN	0,1054
Port 22	N	log diff edConv1D	0,08843
Port 23	AEw	mean std e1d1LN	0,0027
Port 80	AEw	log diff LSTM	0,2763
Port 445	AE	log diff e2d2	3,1e-12
TCP	E	log diff e1d1	0,04933
SSH	Aw	log diff e1d1	0,5037
ICMP	Aw	log GRU	0,03657
MS WBT Server	Ew	mean std e1d1	0,0907
Telnet	AE	log e1d1LN	0,0105

Tabuľka 8: Výsledky Diebol-Marino testu pre jedno krokovú predikciu. Označenia: COU - Počet unikátnych; AP - Iný prístup; A - ARIMA model; E - exponenciálne vyhladzovanie; N - naivný model; AE - ARIMA + exponenciálne vyhladzovanie (priemerovanie); w - metóda posuvného okna.

Tabuľkové hodnoty síce ukazujú v prospech predikcií neurónových sietí oproti štatistickým metódam, no pri bližšom pohľade na hodnoty a hlavne na zobrazenia samotných predikcii je možné vidieť, ze predikcie sú veľmi podobné naivným predikciám. Rovnako, kvôli tomuto problému nevieme prísť k záveru, ktorý typ neurónových sietí a ktorú škálovaciu metódu štandardne voliť. Zobrazenie predikcií umožňuje ich rozdelenie na tri skupiny:

- dobre predikovateľné časové rady,
- nepredikovateľné časové rady,
- nepredikovateľný šum.

Do prvej skupiny časových radov spadá len časový rad vytvorený podľa kritéria Port 443/TCP. Na obrázkoch 10 a 10 sú ukážky najlepších 1 a 10 krokových predikcií tohoto časového radu. Hodnota MASE je v najlepšom prípade jedno krokovej predikcie 0,4209 a v prípade desať krokovej predikcie 0,6025. Port 445/TCP je používaný SMB protokolom na prístup k súborom, tlačiarňam a sériovým portom medzi zariadeniami v

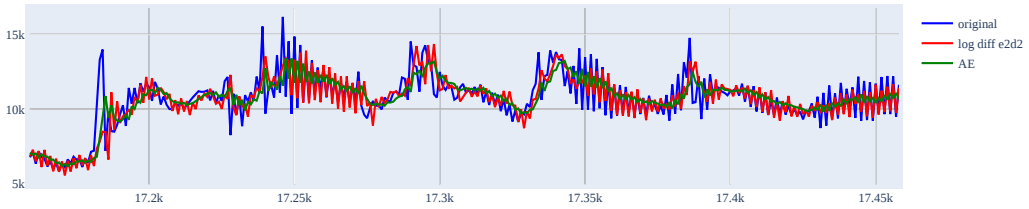
Time series	Stat	Neu	p-value
Count of all alerts	AEw	mean std e1d1	0,0003
COU IPs	Aw	mean std e1d1	0,0012
COU IPs (AP)	AEw	mean std e1d1	0,0024
recon scanning	AEw	log GRU	3,1e-05
attempt login	A	log Conv1DS	0,1580
attempt exploit	Ew	mean std e1d1	0,0422
Port 22	N	log diff edConv1D	0,02685
Port 23	AEw	log e1d1	0,1153
Port 80	E	mean std Conv1DS	0,1974
Port 445	Ew	log diff LSTM	0,6194
TCP	A	log e1d1	0,0959
SSH	Aw	log diff Conv1D	0,2203
ICMP	Aw	mean std e2d2	0,3576
MS WBT Server	AE	mean std Conv1DS	0,1114
Telnet	AE	log e2d2	0,1051

Tabuľka 9: Výsledky Diebol-Marino testu pre desať krokovú predikciu. Označenia: COU - Počet unikátnych; AP - Iný prístup; A - ARIMA model; E - exponenciálne vyhladzovanie; N - naivný model; AE - ARIMA + exponenciálne vyhladzovanie (priemerovanie); w - metóda posuvného okna.

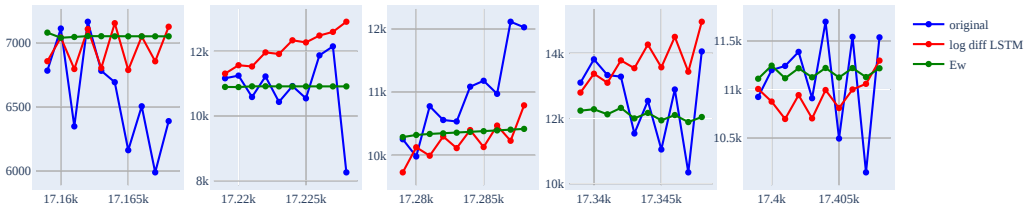
sieti. Je využívaný rôznymi typmi škodlivého kódu (trójske kone, ransomwéry a pod.) na ich šírenie. Zistenie, že tento časový rad je dobre predikovateľný je zaujímave.

V druhej skupine nájdeme až dvanásť časových radov. Ich predikcie sú veľmi podobné naivnej predikcii. Podľa hodnoty MASE ich vieme rozdeliť na dve podskupiny. V prípade prvej podskupiny sa MASE hodnoty pohybujú medzi 0,65 a 0,8 pri jedno krokovej predikcii a pri desať krokovej sú okolo 1 alebo trochu viac ako 1. Podľa ukážok týchto predikcií na obrázkoch 12 a 13 je možné usúdiť, že tieto predikcie sú veľmi blízke naivnej predikcii. Druhá podskupina je na tom ešte horšie. V prípade jedno krokových predikcií sa hodnota MASE pohybuje nad 1 a smerom k desaťkrokovým predikciám ešte rastie. Ukážky predikcií sú na obrázkoch 14 a 15. V tejto skupine nájdeme časové rady vytvorené z portov (22/TCP, 23/TCP, 80/TCP) alebo kategórií (Recon scanning, Attempt login), ktoré sú útočníkmi používané na skenovanie alebo počiatočný prístup do siete. Keďže výsledky predikcii nie sú v tejto skupine dostatočne dobré, aktuálne považujeme túto skupinu za nepredikovateľnú.

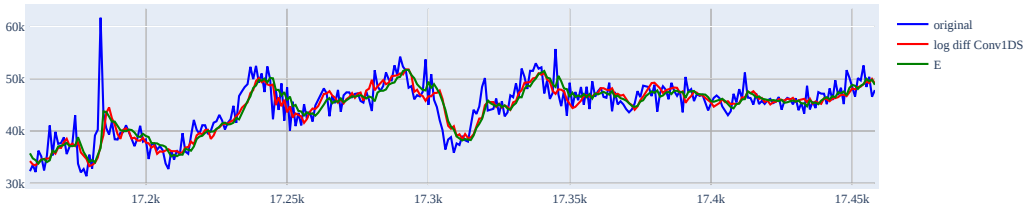
Posledná skupina sa skladá z kategórií Attempt exploit and ICMP protokolu. Ako



Obr. 10: Porovnanie jedno krokovej predikcie časového radu (Port 443) pomocou e2d2 siete s log diff škálovacou metódou a kombinácie metód ARIMA a exponenciálneho vyhladzovania.



Obr. 11: Porovnanie desať krokovej predikcie časového radu (Port 443) pomocou LSTM siete s log diff škálovacou metódou a metódy exponenciálneho vyhladzovania s posuvným oknom.

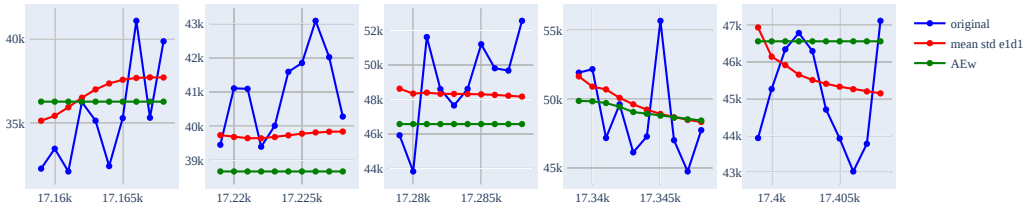


Obr. 12: Porovnanie jedno krokovej predikcie časového radu (Count of all alerts) pomocou Conv1DS siete s log diff škálovacou metódou a kombinácie metód ARIMA a exponenciálneho vyhladzovania.

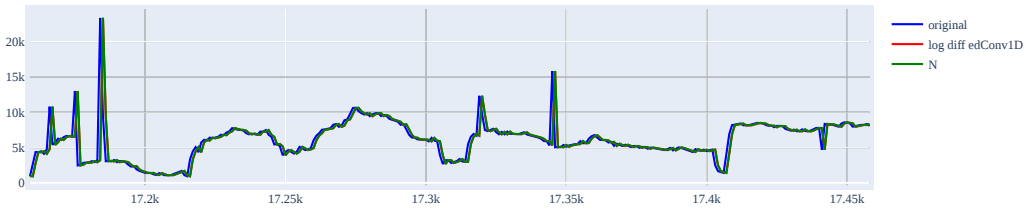
vidieť na obrázkoch 16 a 17 tieto časové rady majú veľmi blízko k bielemu šumu. Zaujímavé sú malé hodnoty MASE (okolo 0,62), ktoré budú spôsobené malým rozdielom medzi predikovanou a reálnou hodnotou.

Výsledky tohoto výskumu nám ukazujú, že enkóder-dekóder siete vyzerajú byť dobrou voľbou pre predikciu v oblasti sieťového bezpečnostného situačného povedomia, no dátová sada, ktorú používame si bude vyžadovať ďalšiu analýzu.

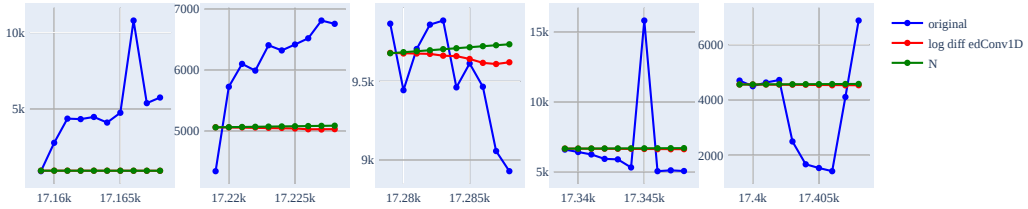
V článku 2 sme sa zamerali na vplyv stratovej funkcie pri tréningu neurónových sietí a výsledky sme opäť porovnali so štatistickými metódami. Použili sme rovnakú dátovú sadu, no len dvojicu časových radov. Jeden z kategórie dobre predikovateľných



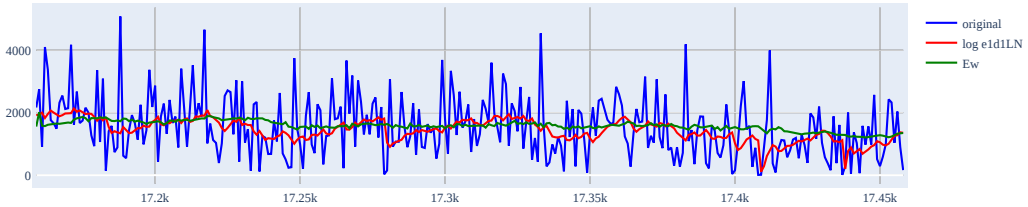
Obr. 13: Porovnanie desať krokovej predikcie časového radu (Count of all alerts) pomocou e1d1 siete s mean std škálovacou metódou a kombinácie metód ARIMA a exponenciálneho vyhladzovania s posuvným oknom.



Obr. 14: Porovnanie jedno krokovej predikcie časového radu (Port 22) pomocou edConv1D siete s log diff škálovacou metódou a naivnej predikcie.

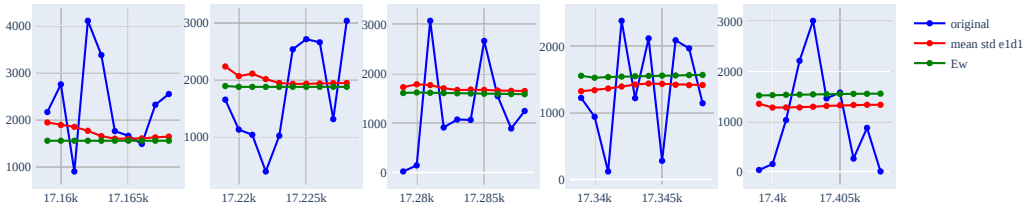


Obr. 15: Porovnanie desať krokovej predikcie časového radu (Port 22) pomocou edConv1D siete s log diff škálovacou metódou a naivnej predikcie.



Obr. 16: Porovnanie jedno krokovej predikcie časového radu (attempt exploit) pomocou e1d1LN siete s log škálovacou metódou a metódy exponenciálneho vyhladzovania s posuvným oknom.

(Port 443/TCP) a druhý z kategórie nepredikovateľných (total number of alerts). Aby sme získali lepší prehľad o kvalite predikcie, zmenili sme rozloženie druhej a tretej časti datasetu tak, aby validačná bola väčšia. Druhá časť obsahovala hodnoty od 15 602 do



Obr. 17: Porovnanie desať krokovej predikcie časového radu (attempt exploit) pomocou e1d1 siete s mean std škálovacou metódou a metódy exponenciálneho vyhladzovania s posuvným oknom.

16 601 a tretia od 16 602 do 17 458. Použili sme rovnaké štatistické metódy ako v predchádzajúcom článku, no z neurónových sietí sme vybrali len päť sieti:

- dopredná sieť so štyrmi skrytými vrstvami (DN),
- rekurentná sieť s troma LSTM vrstvami - užšie vrstvy ako v predchádzajúcom článku (LSTM),
- rekurentná sieť s troma GRU vrstvami (GRU),
- konvolučná sieť s troma 1D konvolučnými vrstvami - s menšími kernelmi ako v predchádzajúcom článku (Conv1D),
- enkóder-dekóder sieť s jednou a jednou LSTM vrstvou (e1d1),

Štatistické metódy boli trénované na zmenšenej druhej časti dátovej sady s a bez metódy posuvného okna a neurónové siete boli trénované na prvej aj druhej časti dátovej sady s väčším časovým oknom 384 hodnôt, teda osem dní. Na rozdiel od minulého článku sme teraz použili aj metódu klesajúceho učiaceho pomeru. Pri tréningu neurónových sietí sme použili tri rôzne stratové funkcie, a to: MAE, MSE a MASE.

Keďže ako jedna z metrík bola použitá funkcia MAE, je dôležité uviesť uviesť štatistické informácie o dátovej sade:

- v prípade časového radu Count of all alerts bola minimálna hodnota 22, maximálna 155 818 a priemer 34 594,25,
- v prípade časového radu Port 445/TCP bola minimálna hodnota 0, maximálna 16 168 a priemer 5 972,65.

V tabuľkách 11 a 12 sú výsledky jednotlivých neurónových sietí pre všetky tri stratové funkcie pre oba časové rady. Ako vidieť, v oboch prípadoch dosahujú MAE a MASE stratové funkcie približne rovnako dobré výsledky a sú lepšie ako v prípade MSE stratovej funkcie. Porovnali sme aj najlepšie výsledky štatistických metód a

neurónových sietí v tabuľke 10. Opäť sa ukazuje, že podľa MASE (aj MAE) metrík dosahujú neurónové siete lepšie výsledky ako štatistické metódy. Pre prípad dvoch časových radov v tomto článku horšie predikcie potvrdené aj Diebold-Marino testom takmer vo všetkých prípadoch.

časový rad	Count of all alerts		port 445/TCP	
najlepšia NN/štat. model	e1d1 MAE	E	GRU MASE	AE
MASE	**0,9166**	1,0319	**0,6210**	0,7661
MAE	**2 437,3081**	2 744,0770	**1 056,5102**	1 303,3641

Tabuľka 10: Porovnanie najlepších neurónových sietí a štatistických modelov na oboch časových radoch. Označenia: NN - neurónová sieť, E - exponenciálne vyhladzovanie; AE - ARIMA + exponenciálne vyhladzovanie (priemerovanie).

metrika	stratová funkcia	DN	LSTM	GRU	e1d1	Conv1D
MASE	MAE	**0,9950**	0,9213	**0,9286**	**0,9166**	**0,9254**
	MSE	1,0245	0,9442	0,9550	0,9430	0,9567
	MASE	1,0147	**0,9192**	0,9352	0,9178	0,9362
MAE	MAE	**2 645,9203**	2 449,9389	**2 469,3886**	**2 437,3081**	**2 460,6580**
	MSE	2 724,2048	2 510,7505	2 539,4539	2 507,7080	2 543,9238
	MASE	2 698,3200	**2 444,1826**	2 486,8879	2 440,6539	2 489,3861

Tabuľka 11: Porovnanie MASE a MAE pre tri stratové funkcie pre všetky predikcie neurónových sietí na časovom rade Count of all alerts na testovacom datasete. Hrubým písmom sú označené najlepšie výsledky každej neurónovej siete.

metrika	stratová funkcia	DN	LSTM	GRU	e1d1	Conv1D
MASE	MAE	**0,6972**	0,6633	0,6307	**0,6408**	**0,7080**
	MSE	0,7215	0,7118	0,7321	0,7038	0,8201
	MASE	0,7020	**0,6617**	**0,6210**	0,6582	0,7208
MAE	MAE	**1 186,1808**	1 128,4426	1 072,9371	**1 090,1418**	**1 204,4519**
	MSE	1 227,4236	1210,9351	1 245,5377	1 197,3586	1 395,2064
	MASE	1 200,1366	**1 125,6894**	**1056,5102**	1 119,8305	1 226,3334

Tabuľka 12: Porovnanie MASE a MAE pre tri stratové funkcie pre všetky predikcie neurónových sietí na časovom rade port 445/TCP na testovacom datasete. Hrubým písmom sú označené najlepšie výsledky každej neurónovej siete.

Záver

V práci sme sa zamerali na predikciu v oblasti kybernetickej bezpečnosti. Konkrétne sa venujeme predikcií sieťovej bezpečnostnej situácie pomocou časových radov. V tejto oblasti sme špecifikovali dvojicu okruhov výskumných problémov. Prvým je chýbajúca benchmarková dátová sada v tejto oblasti a druhým je nutnosť zlepšenia predikčných metód v tejto oblasti. Pre prvý problém sme navrhli, ako by mala potrebná dátová sada vyzerať a popísali sme spôsoby jej vytvorenia. Aktuálne už máme vytvorenú dátovú sadu z dát zo systému WARDEN a ďalšie plánujeme vytvoriť a v budúcnosti zverejniť. V prípade druhého problému sme navrhli štyri metódy, ktorými sa chceme pokúsiť zlepšiť predikciu v sieťového bezpečnostného situačného povedomia. Konkrétne chceme zlepšiť predikciu metódami skladania modelov a augumentácie dát, pomocou pridania ďalších externých údajov na vstup modelom a otestovaním neštandardných stratových funkcií pri tréningu neurónových sietí.

V poslednej kapitole tejto práce sa venujeme výsledkom, ktoré už čiastočne napĺňajú ciele, ktoré sme stanovili v tejto práci. Tie boli publikované v dvoch prácach, ktoré sú uvedené v prílohe tejto práce. V oboch prácach pracujeme s časovými radmi, ktoré boli vytvorené z dát zo systému WARDEN. V prvej sa venujeme predikcii 15 časových radov z oblasti sieťového bezpečnostného situačného povedomia. Súčasne porovnávame výsledky štyroch štatistických modelov (naivného, ARIMA, exponenciálne vyhladzovanie a ich kombinácia) a deviatich neurónových sietí s troma rôznymi metódami škálovania dát. V druhej sa sústreďujeme na vybranú dvojicu časových radov a opäť porovnávame predikcie štyroch štatistických modelov (naivného, ARIMA, exponenciálne vyhladzovanie a ich kombinácia) a piatich neurónových sietí s troma rôznymi stratovými funkciami.

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] Bakdash, J. Z., Hutchinson, S., Zaroukian, E. G., Marusich, L. R., Thirumuruganathan, S., Sample, C., Hoffman, B., and Das, G. Malware in the future? forecasting of analyst detection of cyber events. *Journal of Cybersecurity 4*, 1 (2018), tyy007.

[4] Bandara, K., Hewamalage, H., Liu, Y.-H., Kang, Y., and Bergmeir, C. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognition 120* (2021), 108148.

[5] Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, B., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., et al. Neural forecasting: Introduction and literature overview. *arXiv preprint arXiv:2004.10240* (2020).

[6] Bilgil, H. New grey forecasting model with its application and computer code. *AIMS Mathematics 6*, 2 (2021), 1497–1514.

[7] Borovykh, A., Bohte, S., and Oosterlee, C. W. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691* (2017).

[8] Bou-Harb, E., Debbabi, M., and Assi, C. Cyber scanning: a comprehensive survey. *Ieee communications surveys & tutorials 16*, 3 (2013), 1496–1519.

[9] Byers, R., Waltermire, D., and Turner, C. National vulnerability database (nvd) metadata submission guidelines for common vulnerabilities and expo-

sures (cve) numbering authorities (cnas) and authorized data publishers. Tech. rep., National Institute of Standards and Technology, 2020.

[10] Chatzipoulidis, A., Michalopoulos, D., and Mavridis, I. Information infrastructure risk prediction through platform vulnerability analysis. *Journal of Systems and Software 106* (2015), 28–41.

[11] Chatzoglou, E., Kambourakis, G., and Kolias, C. Empirical evaluation of attacks against ieee 802.11 enterprise networks: The awid3 dataset. *IEEE Access 9* (2021), 34188–34205.

[12] Chen, F., Shen, Y., Zhang, G., and Liu, X. The network security situation predicting technology based on the small-world echo state network. In *2013 IEEE 4th International Conference on Software Engineering and Service Science* (2013), IEEE, pp. 377–380.

[13] Chen, X., Yu, R., Ullah, S., Wu, D., Li, Z., Li, Q., Qi, H., Liu, J., Liu, M., and Zhang, Y. A novel loss function of deep learning in wind speed forecasting. *Energy 238* (2022), 121808.

[14] Cisco. Cisco ios netflow. `https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html`. [Online; accessed 25-May-2022].

[15] Cisco. The history and future of internet traffic. Dostupné na internete: `https://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic`, 2015. [cit. 20. 5. 2022].

[16] Cortez, P., Rio, M., Rocha, M., and Sousa, P. Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems 29*, 2 (2012), 143–155.

[17] Dang, Y., Chen, Z., Li, H., and Shu, H. A comparative study of non-deep learning, deep learning, and ensemble learning methods for sunspot number prediction. *Applied Artificial Intelligence 36*, 1 (2022), 2074129.

[18] Demir, S., Mincev, K., Kok, K., and Paterakis, N. G. Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting. *Applied Energy 304* (2021), 117695.

[19] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (2009), Ieee, pp. 248–255.

[20] Dhiman, H. S., Deb, D., and Guerrero, J. M. Hybrid machine intelligent svr variants for wind forecasting and ramp events. *Renewable and Sustainable Energy Reviews 108* (2019), 369–379.

[21] Diebold, F. X., and Mariano, R. S. Comparing predictive accuracy. *Journal of Business & economic statistics 20*, 1 (2002), 134–144.

[22] Du, S., Li, T., Yang, Y., and Horng, S.-J. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing 388* (2020), 269–279.

[23] 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.

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

[25] Fang, X., Xu, M., Xu, S., and Zhao, P. A deep learning framework for predicting cyber attacks rates. *EURASIP Journal on Information security 2019*, 1 (2019), 1–11.

[26] Fei, H., and He, J. Prediction model of network security situation based on elman neural network [c]. In *The 7th International Conference on Computer Engineering and Networks. SISSA Medialab* (2017), vol. 299, p. 014.

[27] Feng, W., Wu, Y., and Fan, Y. A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit. *International Journal of Intelligent Computing and Cybernetics 13*, 1 (2020), 25–39.

[28] FiRST. Common vulnerability scoring system. Dostupné na internete: `https://www.first.org/cvss/`, 2019. [cit. 20. 5. 2022].

[29] Garcia, S., Grill, M., Stiborek, J., and Zunino, A. An empirical comparison of botnet detection methods. *computers & security 45* (2014), 100–123.

[30] Gheyas, I. A., and Smith, L. S. A novel neural network ensemble architecture for time series forecasting. *Neurocomputing 74*, 18 (2011), 3855–3864.

[31] Gurnani, M., Korke, Y., Shah, P., Udmale, S., Sambhe, V., and Bhirud, S. Forecasting of sales by using fusion of machine learning techniques. In

2017 International Conference on Data Management, Analytics and Innovation (ICDMAI) (2017), IEEE, pp. 93–101.

[32] He, F., Zhang, Y., Liu, D., Dong, Y., Liu, C., and Wu, C. Mixed wavelet-based neural network model for cyber security situation prediction using modwt and hurst exponent analysis. In *International Conference on Network and System Security* (2017), Springer, pp. 99–111.

[33] Hofstede, R., Hendriks, L., Sperotto, A., and Pras, A. Ssh compromise detection using netflow/ipfix. *ACM SIGCOMM computer communication review 44*, 5 (2014), 20–26.

[34] Hoque, M. S., Jamil, N., Amin, N., and Lam, K.-Y. An improved vulnerability exploitation prediction model with novel cost function and custom trained word vector embedding. *Sensors 21*, 12 (2021), 4220.

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

[36] 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.

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

[38] Institute of Distributed Systems, U. U. 2017-suee-data-set. Dostupné na internete: `https://github.com/vs-uulm/2017-SUEE-data-set`, 2017. [cit. 20. 5. 2022].

[39] 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).

[40] Karabiber, O. A., and Xydis, G. Electricity price forecasting in the danish day-ahead market using the tbats, ann and arima methods. *Energies 12*, 5 (2019), 928.

[41] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems 30* (2017).

[42] Kolias, C., Kambourakis, G., Stavrou, A., and Gritzalis, S. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys Tutorials 18*, 1 (2016), 184–208.

[43] Kotu, V., and Deshpande, B. Chapter 2 - data science process. In *Data Science (Second Edition)*, V. Kotu and B. Deshpande, Eds., second edition ed. Morgan Kaufmann, 2019, pp. 19–37.

[44] Lab, S. Hornet: Network dataset of geographically placed honeypots. Dostupné na internete: `https://www.stratosphereips.org/hornet-network-dataset-of-geographically-placed-honeypots`, 2021. [cit. 20. 5. 2022].

[45] Lai, J.-B., Wang, H.-Q., Liu, X.-W., Liang, Y., Zheng, R.-J., and Zhao, G.-S. Wnn-based network security situation quantitative prediction method and its optimization. *Journal of computer science and technology 23*, 2 (2008), 222–230.

[46] Last, D. Using historical software vulnerability data to forecast future vulnerabilities. In *2015 Resilience Week (RWS)* (2015), IEEE, pp. 1–7.

[47] 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.

[48] Leverett, É., Rhode, M., and Wedgbury, A. Vulnerability forecasting: theory and practice. *Digital Threats: Research and Practice* (2022).

[49] Li, R., Li, F., Wu, C., and Song, J. Research on vehicle network security situation prediction based on improved clpso-rbf. In *Journal of Physics: Conference Series* (2021), vol. 1757, IOP Publishing, p. 012148.

[50] Liao, Y., and Liang, C. A temperature time series forecasting model based on deepar. In *2021 7th International Conference on Computer and Communications (ICCC)* (2021), IEEE, pp. 1588–1593.

[51] Lim, B., Arik, S. O., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363* (2019).

[52] Lin, Z., Chen, G., Guo, W., and Liu, Y. Pso-bpnn-based prediction of network security situation. In *2008 3rd International Conference on Innovative Computing Information and Control* (2008), IEEE, pp. 37–37.

[53] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., and Therón, R. Ugr '16: A new dataset for the evaluation of cyclostationarity-based network idss. *Computers & Security 73* (2018), 411–424.

[54] Madan, R., and Mangipudi, P. S. Predicting computer network traffic: a time series forecasting approach using dwt, arima and rnn. In *2018 Eleventh International Conference on Contemporary Computing (IC3)* (2018), IEEE, pp. 1–5.

[55] Moustafa, N. The unsw-nb15 dataset. Dostupné na internete: `https://research.unsw.edu.au/projects/unsw-nb15-dataset`, 2015. [cit. 20. 5. 2022].

[56] Moustafa, N. The ton_iot datasets. Dostupné na internete: `https://research.unsw.edu.au/projects/toniot-datasets`, 2020. [cit. 20. 5. 2022].

[57] Navruzov, Fedor and Halytskyi, Vladyslav. Seq2seq models for time-series forecasting with tensorflow. `https://docs.google.com/presentation/d/1EK4MYilx8RfwRCfbPPolMYBnS8oQvBF9Ddb0B0ThJVs`, 2019. slide 12.

[58] Netresec. 4sics. Dostupné na internete: `https://www.netresec.com/?page=PCAP4SICS`, 2015. [cit. 20. 5. 2022].

[59] of New Brunswick, U. Cicddos2019. Dostupné na internete: `https://www.unb.ca/cic/datasets/ddos-2019.html`, 2018. [cit. 20. 5. 2022].

[60] of New Brunswick, U. Cse-cic-ids2018 on aws. Dostupné na internete: `https://www.unb.ca/cic/datasets/ids-2018.html`, 2018. [cit. 20. 5. 2022].

[61] of Victoria, U. Isot cloud intrusion detection (isot cid) dataset. Dostupné na internete: `https://www.uvic.ca/ecs/ece/isot/datasets/cloud-security/index.php`. [cit. 20. 5. 2022].

[62] Papagiannaki, K., Taft, N., Zhang, Z.-L., and Diot, C. Long-term forecasting of internet backbone traffic. *IEEE transactions on neural networks 16*, 5 (2005), 1110–1124.

[63] Passeri, P. Hackmageddon. Dostupné na internete: `https://www.hackmageddon.com/about/`, 2011. [cit. 20. 5. 2022].

[64] Pekarčík, P., Gajdoš, A., and Sokol, P. Forecasting security alerts based on time series. In *International Conference on Hybrid Artificial Intelligence Systems* (2020), Springer, pp. 546–557.

[65] Pekarčík, P. Spracovania kybernetických bezpečnostných údajov v reálnom čase. Dostupné na internete: https://opac.crzp.sk/?fn=detailBiblioForm&sid=60B4917E3BAC23F10D76A727651C, 2020. [cit. 20. 5. 2022].

[66] Pokhrel, N. R., Rodrigo, H., Tsokos, C. P., et al. Cybersecurity: Time series predictive modeling of vulnerabilities of desktop operating system using linear and non-linear approach. *Journal of Information Security 8*, 04 (2017), 362.

[67] Project, H. Know your enemy: Statistics. Dostupné na internete: https://honeynet.onofri.org/papers/stats/, 2001. [cit. 20. 5. 2022].

[68] Qi, Y., Shang, W., and He, X. A combined prediction method of industrial internet security situation based on time series. In *Proceedings of the 2019 the 9th International Conference on Communication and Network Security* (2019), pp. 84–91.

[69] 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.

[70] Ren, X.-y., Luo, Q.-q., Shi, C., and Huang, J.-m. Network security posture prediction based on sapso-elman neural networks. In *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)* (2020), IEEE, pp. 533–537.

[71] Ring, M., Wunderlich, S., Grüdl, D., Landes, D., and Hotho, A. Creation of flow-based data sets for intrusion detection. *Journal of Information Warfare 16*, 4 (2017), 41–54.

[72] Ring, M., Wunderlich, S., Grüdl, D., Landes, D., and Hotho, A. Flow-based benchmark data sets for intrusion detection. In *Proceedings of the 16th European Conference on Cyber Warfare and Security. ACPI* (2017), pp. 361–369.

[73] Roumani, Y., Nwankpa, J. K., and Roumani, Y. F. Time series modeling of vulnerabilities. *Computers & Security 51* (2015), 32–40.

[74] Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting 36*, 3 (2020), 1181–1191.

[75] Security, D. D. Dds dataset collection. Dostupné na internete: https://datadrivensecurity.info/blog/pages/dds-dataset-collection.html, 2013. [cit. 20. 5. 2022].

[76] Sewdien, V., Preece, R., Torres, J. R., Rakhshani, E., and van der Meijden, M. Assessment of critical parameters for artificial neural networks based short-term wind generation forecasting. *Renewable Energy 161* (2020), 878–892.

[77] Shang, L., Zhao, W., Zhang, J., Fu, Q., Zhao, Q., and Yang, Y. Network security situation prediction based on long short-term memory network. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (2019), IEEE, pp. 1–4.

[78] Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp 1* (2018), 108–116.

[79] Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers & security 31*, 3 (2012), 357–374.

[80] Shu, K., Sliva, A., Sampson, J., and Liu, H. Understanding cyber attack behaviors with sentiment information on social media. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation* (2018), Springer, pp. 377–388.

[81] Spitzner, L. Honeypots: Catching the insider threat. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.* (2003), IEEE, pp. 170–179.

[82] Sun, S., Wei, Y., and Wang, S. Adaboost-lstm ensemble learning for financial time series forecasting. In *International Conference on Computational Science* (2018), Springer, pp. 590–597.

[83] Tang, M., Alazab, M., and Luo, Y. Exploiting vulnerability disclosures: Statistical framework and case study. In *2016 Cybersecurity and Cyberforensics Conference (CCC)* (2016), IEEE, pp. 117–122.

[84] Tang, M., Alazab, M., and Luo, Y. Big data for cybersecurity: Vulnerability disclosure trends and dependencies. *IEEE Transactions on Big Data 5*, 3 (2017), 317–329.

[85] Teixeira, M., Zolanvari, M., and Jain, R. Wustl-iiot-2018, 2020.

[86] telekom security. T-pot - the all in one honeypot platform. Dostupné na internete: `https://github.com/telekom-security/tpotce`, 2022. [cit. 20. 5. 2022].

[87] Turcotte, M. J. M., Kent, A. D., and Hash, C. *Unified Host and Network Data Set*. World Scientific, nov 2018, ch. Chapter 1, pp. 1–22.

[88] University of California, I. Kdd cup 1999 data. Dostupné na internete: `https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, 1999. [cit. 20. 5. 2022].

[89] Vagropoulos, S. I., Chouliaras, G., Kardakos, E. G., Simoglou, C. K., and Bakirtzis, A. G. Comparison of sarimax, sarima, modified sarima and ann-based models for short-term pv generation forecasting. In *2016 IEEE International Energy Conference (ENERGYCON)* (2016), IEEE, pp. 1–6.

[90] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[91] Wang, Z. G., Hu, C. Z., Hao, M. L., and Lu, F. Cyber security datasets research. In *Advanced Materials Research* (2013), vol. 659, Trans Tech Publ, pp. 191–195.

[92] WARDEN. Architektura systému. Dostupné na internete: `https://warden.cesnet.cz/cs/architecture`, 2015. [cit. 20. 5. 2022].

[93] Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478* (2020).

[94] Werner, G., Yang, S., and McConky, K. Time series forecasting of cyber attack intensity. In *Proceedings of the 12th Annual Conference on cyber and information security research* (2017), pp. 1–3.

[95] Werner, G., Yang, S., and McConky, K. Leveraging intra-day temporal variations to predict daily cyberattack activity. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)* (2018), IEEE, pp. 58–63.

[96] Wikipedia. Internet traffic. Dostupné na internete: https://en.wikipedia.org/wiki/Internet_traffic, 2022. [cit. 20. 5. 2022].

[97] Wikipedia contributors. Intrusion detection system — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Intrusion_detection_system&oldid=1079730884, 2022. [Online; accessed 25-May-2022].

[98] world bank, T. Crossing borders. Dostupné na internete: https://wdr2021.worldbank.org/stories/crossing-borders/, 2021. [cit. 20. 5. 2022].

[99] Wu, B., Wang, L., and Zeng, Y.-R. Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy 252* (2022), 123990.

[100] Xiao, P., Xian, M., and Wang, H. Network security situation prediction method based on mea-bp. In *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)* (2017), IEEE, pp. 1–5.

[101] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908* (2020).

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

[103] yd1008. A comparative study of non-deep learning, deep learning, and ensemble learning methods for sunspot number prediction. Dostupné na internete: https://github.com/yd1008/ts_ensemble_sunspot, 2021. [cit. 20. 5. 2022].

[104] Zhan, Z., Xu, M., and Xu, S. Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security 10*, 8 (2015), 1666–1677.

[105] Zhang, G. P., and Berardi, V. L. Time series forecasting with neural network ensembles: an application for exchange rate prediction. *Journal of the operational research society 52*, 6 (2001), 652–664.

[106] Zhang, H., Huang, Q., Li, F., and Zhu, J. A network security situation prediction model based on wavelet neural network with optimized parameters. *Digital Communications and Networks 2*, 3 (2016), 139–144.

[107] Zhang, Y., Jin, S., Cui, X., Yin, X., and Pang, Y. Network security situation prediction based on bp and rbf neural network. In *International Conference on Trustworthy Computing and Services* (2012), Springer, pp. 659–665.

[108] Zhang, Y., Zhu, C., and Wang, Q. Lightgbm-based model for metro passenger volume forecasting. *IET Intelligent Transport Systems 14*, 13 (2020), 1815–1823.

[109] Zheng, R., Zhang, D., Wu, Q., Zhang, M., and Yang, C. A strategy of network security situation autonomic awareness. In *International Conference on Network Computing and Information Security* (2012), Springer, pp. 632–639.

[110] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI* (2021).

Prílohy

V prílohách sa nachádzajú nasledujúce položky:

Príloha A: Zoznam publikácii

Príloha B: Network security situation awareness forecasting based on statistical approach and neural networks

Príloha C: Network security situation awareness forecasting based on neural networks

A Zoznam publikácií autora

1. 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).

2. Staňa, R., Pekarčík, P., Gajdoš, A., and Sokol, P. Network security situation awareness forecasting based on neural networks. In *7th International Conference on Time Series and Forecasting* Springer (2021). [accepted].

3. Vozáriková, G., Staňa, R., Semanišin, G. Clothing parsing using extended u-net. In *VISIGRAPP (5: VISAPP)* (2021), pp. 15–24.

4. Wawrek, I., and Staňa, R. A method of shaping an energy-efficient building envelope based on natural patterns in order to achieve save of energy. In *IOP Conference Series: Materials Science and Engineering* (2021), vol. 1209, IOP Publishing, p. 012071.

5. Andrejková, G., Antoni, L., Bruoth, E., Bugata, P., Gajdoš, D., Guniš, J., Horvát, Š., Hudák, D., Kmečová, V., Krajči, S., et al. Applications of deep learning models within national project it academy–education for 21st century. In *2021 19th International Conference on Emerging eLearning Technologies and Applications (ICETA)* (2021), IEEE, pp. 12–17.

6. Antoni, L., Bruoth, E., Bugata, P., Bugata Jr, P., Gajdos, D., Horvat, S., Hudak, D., Kmecova, V., Stana, R., Stankova, M., et al. Automatic ecg classification and label quality in training data. *Physiological Measurement* (2022).

7. Bruoth, E., Bugata, P., Gajdoš, D., Horvát, Š., Hudák, D., Kmečová, V., Staňa, R., Staňková, M., Szabari, A., Vozáriková, G., et al. A two-phase multilabel ecg classification using one-dimensional convolutional neural network and modified labels. In*2021 Computing in Cardiology (CinC)* (2021), vol. 48, IEEE, pp. 1–4.

Logic Journal of the IGPL, 2021, pp. 1–13

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

Network security situation awareness forecasting based on statistical approach and neural networks

Pavol Sokol,[1,*] Richard Staňa,[1] Andrej Gajdoš[2] and Patrik Pekarčík[1]

[1]Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, 04001, Košice, Slovakia and
[2]Institute of Mathematics, Faculty of Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, 04001, Košice, Slovakia
*Corresponding author. pavol.sokol@upjs.sk
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

The usage of new and progressive technologies brings with it new types of security threats and security incidents. Their number is constantly growing.The current trend is to move from reactive to proactive activities. For this reason, the organisation should be aware of the current security situation, including the forecasting of the future state. The main goal of organisations, especially their security operation centres, is to handle events, identify potential security incidents, and effectively forecast the network security situation awareness. In this paper, we focus on increasing the efficiency of utilisation of this part of cybersecurity. The paper's main aim is to compare selected statistical models and models based on neural networks to find out which models are more suitable for network security situation awareness forecasting. Based on the analysis provided in this paper, neural network methods prove a more accurate alternative than classical statistical prediction models in network security situation awareness forecasting. In addition, the paper analyses the selection criteria and suitability of time series, which do not only reflect information about the total number of security events but represent a category of security event (e.g. recon scanning), port or protocol.

Key words: Neural networks, Forecasting, Network situational awareness, Time series

Introduction

The number of cyber threats and attacks targeted towards all varieties of devices increases daily. The main topics in cybersecurity are the detection of cybersecurity incidents and the response to them. Security threats cannot be completely eliminated. Therefore, the current trend is to move from reactive to proactive activities [8]. The main goal is to prevent or mitigate security incidents before they cause harm to the organisation.

From the point of view of the organisation, it is vital to know the current situational awareness, which can be characterised as *"Perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future"* [14]. In the area of cyber threats and cybersecurity incidents, we consider the network security situation awareness (NSSA). In [2], Bass firstly introduced this concept, and it can be defined as *"a monitoring of cyber systems, understanding of the cybersecurity situation represented by modelling of cyber threats or relating security alerts and predicting the changes in cybersecurity situation"* [23].

Bass divided the network security situation awareness (NSSA) in paper [2] to three stages:

- event detection - this stage identifies the abnormal and malicious activities in the network and obtains the basic information and some important factors related to the network security,
- current situation assessment - this stage analyses collected data and evaluate the network security situation by using the information obtained from the previous stage,
- future situation prediction - this stage aims to forecast the future network security tendency.

The main objective of the NSSA is the last stage - future situation prediction. In general, predictive analysis methods play a significant role in predicting specific security incidents, predicting the next steps of the attacker or predicting the organisation's security situation. In this regard, we

recognise three main approaches to predictive methods in cybersecurity [23]:

- attack projection - predicting the next move of an adversary in a running attack by projecting the series of actions the attacker performs [50];
- attack prediction - a type of attacks are going to happen where and when [1];
- security situation forecast - forecast number of attacks or vulnerabilities in the network of the organisation [33].

As the authors stated in the paper [23], the improvements of prediction in cyber security, including the NSSA forecasting, are a very important issue. From this point of view, the authors identified an important aspect - the utilisation of NSSA forecasting in practice. As mentioned above, the current trend is to move from reactive to proactive activities. The main goal of security operation centres is to handle events, identify potential security incidents, and effectively forecast the NSSA. There are several issues associated with this. The utilisation of NSSA forecasting means considering the forecasting methods' space and time requirements, criteria for suitable forecasting methods and criteria for suitable time series of security alerts, etc. It can help with more effective NSSA forecasting.

To summarise the problems outlined above, we emphasise the following research issues that we aim to answer:

1. comparison of selected models and data scaling for finding out which is more suitable for NSSA forecasting;
2. time series selection criteria suitable for NSSA forecasting concerning statistical methods and neural networks.

Whereas the predominant forecasting methods in cyber security research and operation are statistical methods and methods based on neural networks [23], we have decided to forecast time series by these methods to answer research questions. Time series models "attempt to make use of the time-dependent structure present in a set of observations " [10]. The appropriate forecasting methods depend primarily on what type of data is available. We choose either qualitative forecasting methods (in cases when available data are not relevant to the forecasts) or quantitative forecasting methods. For the purpose of research in this paper, we have available data from security data-sharing platform called Warden system [29] and we have chosen quantitative forecasting methods, which describe the NSSA at a point in time [33].

This paper is an extension of our previous research results [40, 22]. In the earlier papers, we focused on the usage of time series in the NSSA forecasting. In the published paper, we applied statistical methods for forecasting time series. The main goal of this paper was to determine the effect of seasonality and sliding window on NSSA forecasting and criteria for choosing the suitable time series. In this paper, we bring neural network models to our research.

This paper is organised into six sections. Section 2 focuses on the review of published research related to NSSA forecasting based on time series and neural networks. Section 3 focuses on the research methodology and outlines the dataset and methods used for the analysis. Section 4 discusses an experiment evaluation. Section 5 states the result from analysing the research questions and discussing knowledge obtained from the analysis. The last section contains conclusions and future works.

Related works

This section provides an overview of papers related to NSSA forecasting using time series analysis or neural networks.

A frequently used class of models in the area of NSSA forecasting is the Auto-Regressive Integrated Moving Average (ARIMA). Examples of research work using these forecasting methods are [38, 47, 48] In the paper [38], the author uses a broad range of unconventional signals from various public and private data sources and a set of signals forecasted via the ARIMA model. Authors in [47] aim to exploit temporal correlations between the number of attacks (denial of service, malicious email, malicious URL, and attack on internet-facing service) per day to predict the future intensity of cyber incidents. On the other hand, authors in [48] investigated the use of ARIMA models to forecast daily counts of different cyberattack types against multiple targets. The attack methods include in the dataset are malware, malicious URL, and malicious email.

ARIMA models are often used in combination with other models. In the paper [46] authors used ARIMA models and Bayesian Networks to predict future cyber attack (malware, malicious URL, and malicious email) occurrences and intensities against two target entities. ARIMA models are also used in the NSSA forecasting with the gray-box prediction that uses gray-box models that can accommodate the statistical properties exhibited by the data. The authors of the paper [42] responded to the disadvantages of the different use of the ARIMA model (strict requirements on the input) and the gray-box model (based on the prediction of the exponential rate and does not take into account the randomness of the system). A combined prediction model can be constructed to make up for the shortcomings of the two single models. This analysis was shown on industry data (isopropanol refining process). Another example of the use of gray-box models is paper [51]. The authors analysed the extreme-value phenomenon, which means the models' incapability partly causes the prediction errors to predict significant attack rates. The authors used two complementary approaches - the Extreme Value Theory and the Time Series Theory. The authors show that Extreme Value Theory can offer long-term predictions (24-hour ahead-of-time) in this paper. At the same time, gray-box time series forecasting models (FARIMA + GARCH models) can predict attack rates 1-hour ahead of time.

For NSSA forecasting purposes, ARMA models are also used in combination with another method. The objective of the research in the paper [41] is an analysis of the fitting of ARMA and GARMA models to the cyber-attack process. The authors found that the performance of the GARMA model in cyber-attack is perfect compare to ARMA. The authors used NetFlow data obtained from communication on the university honeynet for the analysis. In the paper [49] authors reported a statistical analysis of a breach incident data set corresponding to 12 years of cyber hacking activities that include malware attacks. The AR (1) model can also be used for NSSA forecasting. In the paper [43] authors used model AR (1) and bootstrap based on AR (1) model to forecast NSSA. For this purpose, they used medium-interaction honeypots (collected SSH connections).

Neural networks are commonly used in the field of time series prediction. In cybersecurity, there is plenty of papers that use older types of smaller feed-forward networks or wavelet neural networks trained by backpropagation or genetics algorithm (and its variants) to predict NSSA, for example, [55, 45, 53, 34, 31, 52, 21]. Modern approaches like recurrent neural

networks (GRU, LSTM) were used in [16] for predicting NSSA. In [20], authors compare ARIMA, LSTM and GRU neural networks for cyber attacks prediction from the combination of time series and external signals. Another work [15] predict time series created from data collected by honeypot using a bi-directional LSTM neural network. Research groups are working with recurrent neural networks like LSTM and GRU for predicting cyberattacks from time series created from industrial data [32, 17, 18].

Methodology

Dataset

The source of data for our research is the alerts obtained from a system called Warden [29]. It is a system that supports sharing information about security events on individual computer networks connected to this system. Data is stored and shared in IDEA (Intrusion Detection Extensible Alert) format [28]. IDEA format is a descriptive data model using a key-value JSON structure.

The primary detection data sources that send IDEA alerts to the Warden system include attack detection systems, honeypots, network flow probes, system log records, and other data sources deployed in several networks (Czech national research and education network, Czech commercial network). Alert in the IDEA format contains several mandatory fields (form, ID, detection time, category) [24] and many optional fields with multiple input support. The fields we follow most in this research are the category, network traffic source data (IP address, port, protocol), network traffic target data (IP, port, protocol), detection time, and interruption time. For this research, data were collected during one year (from 2017-12-11 to 2018-12-11). The collected dataset contains approximately one billion records from various data sources mentioned above.

For this research, we processed a one-year dataset to 21-time series based on different criteria. Criteria were chosen from all possible values of category, port and protocol fields by statistical representation throughout the whole dataset. Chosen criteria are followings: (I) Count of all alerts; (II) Count of unique IPs; (III) Category recon scanning; (IV) Category availability DDoS; (V) Category attempt login; (VI) Category attempt exploit; (VII) Category malware ransomware; (VIII) Category intrusion botnet; (IX) Port 21; (X) Port 22; (XI) Port 23; (XII) Port 25; (XIII) Port 80; (XIV) Port 443; (XV) Port 445; (XVI) Protocol TCP; (XVII) Protocol SSH; (XVIII) Protocol UDP; (XIX) Protocol ICMP; (XX) Protocol MS WBT Server; (XXI) Protocol telnet.

Furthermore this time series was processed to multiple versions based on reference period. We made a time series with 1 minute, 10 minutes, 15 minutes, 30 minutes, and 60 minutes. Time series is stored in PostgreSQL database with TimescaleDB extension. TimescaleDB extension is vital for our future aggregations to create different views on created time-series data.

Method Description

There is a wide range of quantitative forecasting methods, and their usage often depends on the specific disciplines, the nature of data or particular purposes. For choosing a particular method, the properties, accuracies, and computational costs must be considered. In our research, we are comparing statistical methods with neural networks for time series forecasting. From statistical approaches, we choose: ARIMA models; Exponential smoothing models (state-space models); the naive method (with drift); combination (average) of ARIMA and Exponential smoothing models and from neural networks, we choose more types of networks: Dense network; Recurrent neural networks (LSTM, GRU); Convolutional neural networks; Encoder-Decoder networks.

Statistical methods

The most commonly used classes of models in time series modelling and forecasting are ARIMA and Exponential smoothing (ETS) [25]. We compared them with the naive methods [5, 4], which can process large data sets and do not have high computational requirements. They serve as a benchmark for predictions in our research. We also added a combination (average) of the ARIMA and ETS methods to compare the standard methods with their combination. The idea of averaging or boosting is trendy in machine learning nowadays [23].

Prediction using ETS family models is characterised by a weighted combination of older observations with new ones. The new observations have a relatively higher weight compared to the older observations. Exponential smoothing reflects the fact that weights decrease exponentially as observations age [25, 5].

The ARIMA models represent a generalisation of the class of ARMA models that incorporate a wide range of non-stationary series. By finite number of differentiations, these models ensure time-series stationarity, allowing the use of ARMA models. ARMA models are a combination of auto-regression (AR) and moving average (MA) [4]. The ETS class provides another approach to time series modelling and forecasting. While ETS models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data [25].

Neural networks

On the other hand, we use nine different neural networks. There is a lot of work done in time series forecasting with neural networks. For example in field of stock prediction [39, 7, 30], traffic prediction [19, 54], etc. We developed more multi-layers neural networks, most of which were inspired by [6]. Because our dataset was quite more extensive than datasets standardly used in literature, we chose slightly bigger architectures of networks so they can be reduced if needed in future work. It is not our goal to do an extensive neural network parameter tuning in this paper. We would like to find out which network type is more suitable for our purposes (recurrent, convolutional or Encoder-Decoder) and we focused on finding out which data scaling is best for our NSSA dataset for neural network training. In the following text, we provide a description of the architectures (abbreviations, used later and denoting individual architectures are in parentheses), N denotes N-steps prediction:

- Dense network (DN) - 4 dense layers (1024, 512, 256, 128 units; activation relu), 1 dense layer (N unit; activation linear);
- Long Short-Term Memory (LSTM) - 3 LSTM layers (512, 512, 512 units; default parameters), 1 dense layer (N units; activation linear);
- Gated Recurrent Unit (GRU) - 3 GRU layers (256, 256, 256 units), 1 SimpleRNN layer (128 units), 1 dense layer (N unit; activation linear);
- 1D convolution (Conv1D) - 3 Conv1D layers (256, 256, 256 filters; 7, 7, 7 kernel size; activation relu; pading same), 1

dense layer (64 unit; activation relu), 1 dense layer (N units; activation linear);
- 1D convolution (Conv1DS) - 3 Conv1D layers (256, 128, 64 filters; 7, 5, 3 kernel size; activation relu; pading same), 1 dense layer (64 unit; activation relu), 1 dense layer (N units; activation linear);
- Encoder-Decoder Conv1D (edConv1D) - 2 Conv1D layers (256, 128 filters; 7, 5 kernel size; activation relu; pading same), 1 dense layer (1 unit; activation sigmoid), 1 dense layer (9216 units; input dimension 128; activation linear), 2 Conv1DTranspose layers (128, 256 units; 5,7 kernel size; activation relu; pading same), 1 dense layer (N units; activation linear);
- Encoder-Decoder LSTM (e1d1) - 1 LSTM encoder layer (512 units encoder; return state True), RepeatVector layer, 1 LSTM decoder layer (512 units encoder; return state True), TimeDistributed (1 dense unit; activation linear);
- Encoder-Decoder LSTM with layer normalization (e1d1LN) - 1 recurrent encoder layer (512 LSTM with layer normalization units; return state True), RepeatVector layer, 1 LSTM decoder layer (512 with layer normalization units units), TimeDistributed (1 dense unit; activation linear);
- Encoder-Decoder LSTM (e2d2) - 2 LSTM encoder layer (512, 512 units encoder; return state True), RepeatVector layer, 2 LSTM decoder layer (512, 512 units encoder), TimeDistributed (1 dense unit; activation linear).

Forecast accuracy evaluation metrics

For forecast accuracy evaluation, the mean absolute scaled error (MASE) is commonly used [27]. It is a preferred metric as it is less sensitive to outliers, more easily interpreted and less variable on small samples. MASE is defined as [25]:

$$\text{MASE} = \text{mean}(|q_j|) \tag{1}$$

where, for non-seasonal time series, q_j is:

$$q_j = \frac{e_j}{\frac{1}{T-1}\sum_{i=2}^{T}|y_i - y_{i-1}|}, \tag{2}$$

where e_j is forecast error, i.e., the difference between an observed value and its forecast, y_j represents observed value, T is the length of time series, and m is seasonality parameter (period).

Experiment evaluation

The research questions were evaluated using a dataset from the aforementioned security data-sharing platform (Warden system). We have divided our evaluation into several stages:

- 1st stage - preliminary stage focused on the best combination of time interval, period, and period in the NSSA forecasting and analysis of the seasonality and the usage of rolling windows;
- 2nd stage - comparison of selected statistic and neural models and data scaling;
- 3rd stage - analysis of NSSA forecasting in order suitable time series selection criteria.

The first stage of evaluation

The first stage of our research focused only on statistical models (ARIMA, ETS, ARIMA+ETS, Naive model). For each class of models, we fitted particular models in the seasonal and non-seasonal settings. We considered one, two, five, and ten steps ahead forecast compared to true values included in the test set.

Furthermore, we considered two cases of model fitting. The first one was the "classical" one; when we kept the training dataset (1200 values) and step by step, we added one more observation from the test set to the training set in each round of evaluation. In the second case, we used the so-called "rolling window" or "one in, one out", which means that in each round of evaluation, we remove the oldest observation from the training set and at the same time we add one new observation from the test set to the training set.

At first, 95 % (bootstrap) prediction intervals were calculated, and consequently, we computed the average coverage. It is the percentage of all confidence intervals which covered the true (future) value of particular time series. We also took a look at the average length of prediction intervals. In general, shorter prediction intervals are considered more precise.

In the first step of this stage, we evaluated forecasting methods only on the total number of alerts. We did not address the qualitative component (alert category, network protocol, or network port). We used a one-year dataset and considered 24 hours and seven days period with two different time units (30 minutes and 60 minutes) and two different lengths of datasets (month, two months). The main aim of this step was to answer the issues of seasonality and the usage of rolling windows in the perspective of NSSA long-term period (one year).

In the second step of this stage, we used the best combination of time interval, period, and forecasting period (30 minutes, seven days, one month). In this step, we evaluated forecasting methods on 21 attributes of alerts, and we addressed the qualitative component (alert category, network protocol, or network port). The main aim of this step was to analyse seasonality and the usage of rolling windows in other time series taking into account qualitative components.

You can find a more detailed description of the experimental evaluation of this stage in our previous work [40].

The second stage of evaluation

In the second stage of the evaluation, we focus on the comparison of prediction models based on statistical approaches and neural networks and finding out which data scaling method is better for neural network training.

We choose 30 minutes time units for the time series. Every time series consist of 17473 values. We did not use the first 27 and last 14 values because there were primarily zeros or missing values.

We used R functions from one of the most common R-packages called *forecast* [26] to implement statistical approaches from the previous stage also described in section Methodology (ARIMA, ETS, ARIMA+ETS and Naive model, all with and without sliding window). Values between 15960 and 17159 (1200 values) were used to fit the models and values between 17160 and 17459 (300 values) to test their forecasting accuracy.

We use a window size of 144 values (3 days) for every model and every time series for neural network training. The last 300 values (as in statistical methods) were used to test the neural networks. The actual size of the testing set was 300 + 144 values, because finally, we wanted to compare all models on these 300 values.

Due to missing values between 15550 and 15601 in the whole dataset, we split the training part of the dataset into two parts. There were values between 28 and 15549 (15522 values) in the first part, and in the second part were values between 15603 and 17159 (1557 values). After preparing data for training first and second parts were concatenated, and training involved 40 epochs.

The fact that we use more data for training neural networks than for fitting statistical models is not problem because older data have weak impact on statistical methods and on the other side it is better for neural networks to have more data to train on. With more data neural networks can be better for finding patterns in time series even in older data.

It is important to scale features before training a neural network. We used three types of dataset scaling for neural networks (abbreviations, used later and denoting scaling are in parentheses):

1. Log difference (log diff) inspired by [37];
2. Natural logarithm (log);
3. Subtraction of mean and division by standard deviation (mean std), mean and standard deviation was calculated using the first part of the training set.

GPU Nvidia GTX 1080 and 1060, TensorFlow version 2.4 were used to train neural networks. Mean squared error (MSE) was employed as loss function, the metric was set to MAE, and the Adam optimiser was used. The batch size was set to 128. We compared all employed models according to MAE for each time series when the training phase is over. So we chose the best model in each scenario. For easier model comparison, we use the tool Weights & Biases [3]. In total, we train more than 1620 neural networks (9 networks described in Methodology x 3 scaling methods x 4-time steps x 15-time series).

In this manner, we obtained predictions (1-step, 2-steps, 5-steps, and 10-steps) by neural networks for each time series. Next, we compared the predictions of neural network models with those acquired by statistical methods based on MASE, as seen in the following section.

Apart from the comparison of prediction methods based on statistical methods and neural networks, we would like to find out what neural network type and what data scaling is most suitable for time series forecasting tasks in our case.

The third stage of evaluation

In the last stage of the evaluation, we analysed the suitability of time series for forecasting. We evaluated 15-time series as in the previous section. For this purpose, we used all forecasting methods and the MASE evaluation criterion. We analysed MASE values of statistical and neural network approaches for each time series.

We conduct this analysis for 1-step, 2-steps, 5-steps and 10-steps forecasts. Subsequently, we selected and visualised the best representative for statistical methods and models based on neural networks. The resulting graphs show an example of the development of values in the time series in contrast to their forecasts.

Results and discussion

In this section, we describe in more detail the stages of experiment evaluation. We take a closer look at the individual results and discuss research questions based on them.

Results of the first stage of the evaluation

In the first step of the first stage, we tested statistical methods (ARIMA, ETS, ARIMA+ETS and Naive model) in six cases (time interval, period, time unit):

- 30 minutes, 24 hours, one month;
- 30 minutes, seven days, one month;
- 60 minutes, 24 hours, two months;
- 60 minutes, 24 hours, one month;
- 60 minutes, seven days, two months;
- 60 minutes, seven days, one month.

The first approach to evaluate the predictions' quality of particular models is the so-called cross-validation [25]. The second approach to assessing the quality of forecasts of specific models is an average coverage of 95 % by prediction intervals. We used the above six cases and compared the established forecasting approaches based on the 1-step, 2-steps, 5-steps, and 10-steps forecast.

The main aim of this step was to determine an appropriate case. According to results mentioned in [40] the most suitable scenario for the methodology chosen by us is the case with monthly data, time intervals of 30 minutes, a period equal to 7 days. The research-based also confirms the suitability of selecting the 7 days on time-oriented analysis and visualisation of data collected by honeypots [44].

In the second step of the first stage, we tested statistical methods (ARIMA, ETS, ARIMA+ETS, Naive model) with the best result case (30 minutes, seven days, one month) on 16 selected time series. Six-time series have been removed from the evaluation process because they contained small counts (less than 100 – e.g. category availability DDoS) or mostly zeros (e.g. category malware – ransomware). If the counts are large enough (more than 100), then the difference between a continuous sample space and the discrete sample space has no perceivable effect on the forecasts [25].

In this step, we calculated the average MASE values for 2-steps, 5-steps forecasts and 10-steps forecasts for 16-time series. The results from this step confirm the findings from the first step of the first stage. The best method seems to be the combination (average) of ARIMA and Exponential smoothing models. Although the methods with sliding window do not have the best results in all cases (16-time series), their results in the average MASE value are close to the best method. More detailed results can be found in the paper [40].

We chose the ETS and ARIMA methods as they can reflect seasonality in the data. We hypothesised that it is unnecessary to take seasonality into account, as the patterns in the data do not repeat regularly (in some cases only irregularly) but depend on other factors. As the results showed, the best values for individual forecasts are always achieved by the method in both its forms - taking into account, respectively, regardless of seasonality. We discuss this in more detail in the paper [40]. For this reason, we do not consider statistical methods that take seasonality into account in the following stages of our evaluation.

Results of the second stage of the evaluation

In this subsection, we discuss the comparison of statistical prediction models and forecasting models based on neural networks and the influence of the data scaling method.

Based on previous work [40] six time series have been removed from the evaluation process because they contained

small counts (less than 100 – e.g. category availability DDoS) or contained mostly zeros (e.g. category malware – ransomware). If the counts are large enough (more than 100), then the difference between a continuous sample space and the discrete sample space has no perceivable effect on the forecasts [25]. However, if our data contains small counts (less than 100), then we need to use forecasting methods that are more appropriate for a sample space of non-negative integers. For instance, the so-called Croston's method can be considered [11, 9]. On the other hand, the time series with mostly zeros can be suitable for time series anomaly detection [36], and it is an exciting direction for future research.

Comparison of neural networks and statistical approaches

All above mentioned statistical forecasting methods (all four methods with and without sliding window) and forecasting based on neural networks are compared by MASE criterion within 1-step, 2-steps, 5-steps, and 10-steps forecasting on 15 selected time series. In total, we compared 11 forecasting methods (eight statistical approaches and best neural network models employing three different data-scaling approaches) per each of 15-time series. In the following text, we discuss all forecasting scenarios in more detail.

Within 1-step forecasting, according to the MASE criterion, forecasting methods based on neural networks are better than statistical approaches for 13 investigated time series. Statistical approaches performing is better in case of attempt login and SSH time series.All three implemented data-scaling approaches have comparable results. As it can be seen in Table 5 the best MASE value is 0.4209 in the case of Port 445 predicted by the e2d2 model with log diff scaling method. An example of prediction can be seen in Figure 1.

According to MASE criterion, in the case of 10-steps forecasting, forecasting methods based on neural networks are better than statistical approaches (except SSH). All three investigated data-scaling methods have comparable results. As shown in Table 6 the best MASE value is 0.6025 in the case of Port 445 predicted by Conv1DS model with log scaling method. An example of prediction can be seen in Figure 2.

According to the MASE criterion, in the case of 2-steps and 5-steps forecasting, forecasting methods based on neural networks are better than statistical approaches in all investigated time series (except SSH in the 5-steps case). All three investigated data-scaling methods have comparable results. We do not present tables for two and 5-steps cases because the values are similar or average between the values of 1 and 10-steps predictions. The best MASE value is again in Port 445 with a value of 0.4712 (e2d2 model with log diff scaling method) for 2-steps and 0.5964 (e1d1LN model with log scaling method) for 5-steps.

As it can be seen in Table 5 and 6, according to MASE, neural network models are better in the most cases. Nevertheless, a lot of cases the best methods are worse or comparable to the naive forecasting with drift (MASE value is greater than or equal to 1). Even if MASE values are much smaller than one most predictions still look like naive predictions or worse as you can see on Figures 3, 4, 5, 6, 7 and 8. This problem is connected to the unpredictability of given time series and will be discussed more precisely in the next stage of the evaluation.

Additionally, we compared the forecast accuracy of the best methods from neural networks and statistical approaches applying the Diebol-Mariano test [12]. We calculated this test using *multDM* [13] package in R. The null hypothesis was that two forecasts have the same accuracy and the alternative hypothesis had a setting H1= "less" (the first forecast is less accurate than the second forecast). We stated that the p-value for confirming the null hypothesis should be higher than 0.05 (we leave a 5% uncertainty rate).

Time series	Stat	Neu	p-value
Count of all alerts	E	log diff Conv1DS	0.0200
COU IPs	E	log diff DN	0.1914
COU IPs (AP)	E	log diff GRU	8.5e-05
recon scanning	Ew	log diff e1d1	0.0110
attempt login	A	log diff GRU	0.6613
attempt exploit	Ew	log e1d1LN	0.1054
Port 22	N	log diff edConv1D	0.08843
Port 23	AEw	mean std e1d1LN	0.0027
Port 80	AEw	log diff LSTM	0.2763
Port 445	AE	log diff e2d2	3.1e-12
TCP	E	log diff e1d1	0.04933
SSH	Aw	log diff e1d1	0.5037
ICMP	Aw	log GRU	0.03657
MS WBT Server	Ew	mean std e1d1	0.0907
Telnet	AE	log e1d1LN	0.0105

Table 1. Results of Diebol-Marino test for 1-step forecasting. Notes: A - ARIMA model; E - Exponential Smoothing (state space models); N - naive model; AE - ARIMA + Exponential smoothing (average); w - rolling window; AP - another approach.

Time series	Stat	Neu	p-value
Count of all alerts	AEw	mean std e1d1	0.0003
COU IPs	Aw	mean std e1d1	0.0012
COU IPs (AP)	AEw	mean std e1d1	0.0024
recon scanning	AEw	log GRU	3.1e-05
attempt login	A	log Conv1DS	0.1580
attempt exploit	Ew	mean std e1d1	0.0422
Port 22	N	log diff edConv1D	0.02685
Port 23	AEw	log e1d1	0.1153
Port 80	E	mean std Conv1DS	0.1974
Port 445	Ew	log diff LSTM	0.6194
TCP	A	log e1d1	0.0959
SSH	Aw	log diff Conv1D	0.2203
ICMP	Aw	mean std e2d2	0.3576
MS WBT Server	AE	mean std Conv1DS	0.1114
Telnet	AE	log e2d2	0.1051

Table 2. Results of Diebol-Marino test for 10-steps forecasting. Notes: A - ARIMA model; E - Exponential Smoothing (state space models); N - naive model; AE - ARIMA + Exponential smoothing (average); w - rolling window; AP - another approach.

Results of Diebold-Marino test of the best statistical method and neural network method for 1-step forecasting are shown in Table 1 and for 10-steps forecasting are shown in Table 2. In these tables, column "Time series" means used time series, the column "Stat" means the best neural methods, and the column "p-value" means p-value for Diebold-Marino test for a couple of the statistic and neural methods.

For example, in the case of the 1-step forecast (Count of all alerts time series), the best statistical method is ETS, and the best neural networks method is Log diff representative. We tested the null hypothesis (forecasts have the same accuracy) against the alternative hypothesis (forecast based on the

		DN	LSTM	GRU	Conv1D	Conv1DS	edConv1D	e1d1	e1d1LN	e2d2
1-step	log diff	1	2	2	0	1	1	3	1	4
	log	0	4	2	2	2	0	1	3	1
	mean std	1	1	3	0	0	2	4	4	0
	sum	2	7	7	2	3	3	8	8	5
2-steps	log diff	0	4	2	0	1	1	4	2	1
	log	0	1	2	4	3	0	2	1	2
	mean std	1	1	1	0	1	2	4	4	1
	sum	1	6	5	4	5	3	10	7	4
5-steps	log diff	0	1	2	0	1	2	4	3	2
	log	0	0	2	2	3	0	5	1	2
	mean std	1	2	1	0	1	3	5	1	1
	sum	1	3	5	2	5	5	14	5	5
10-steps	log diff	0	1	4	1	1	1	2	2	3
	log	0	1	1	1	3	2	3	0	4
	mean std	1	0	0	0	2	4	4	1	3
	sum	1	2	5	2	6	7	9	3	10
in total	log diff	1	8	10	1	4	5	13	8	10
	log	0	5	7	9	11	2	11	5	9
	mean std	4	5	5	0	4	11	17	10	5
	sum	**5**	**18**	**22**	**10**	**19**	**18**	**41**	**23**	**24**

Table 3. Summary of counts selected models based on neural network with the best performance for each scaling method.

statistical method is less accurate than the forecast based on neural network method). Since P-value is 0.03239, which is less than 0.05, the null hypothesis has been rejected, and the alternative hypothesis has been accepted in this case. In other words, the forecast based on ETS is less accurate than a forecast based on the log diff representative method in the case "Count of all alerts" time series.

Table 1 and Table 2 show the results of Diebold-Marino test for the NSSA forecasting. The neural network methods are better than statistical methods in the 8 cases (1-step forecasting) and the 6 cases (10-steps forecasting). In the other cases (time-series), the both best forecasting methods have the same accuracy.

From our experiments, we can conclude that neural network methods have better accurate forecast in some cases for the NSSA forecasting, which is in contradiction with study [35] that evaluated and compared the performance of many classical methods (such as linear methods and exponential smoothing) and modern machine learning and deep learning methods (such as Multilayer Perceptrons, LSTM, etc.) on a large and diverse set of more than 1,000 univariate time series forecasting problems and classical methods did outperform machine learning methods.

Comparison of neural network type and data scaling methods
Since we considered several types of neural networks types and several data scaling methods in our analysis, we compared different combinations of these methods. The aim was to identify suitable types and scaling methods.

Table 3 is the summary of models based on a neural network with the best performance. The last row shows several cases when the neural network method has the best results for all combinations of all-time series, scaling methods and n-steps forecasting. Dense networks were used as a baseline, and we did not expect them to perform well. Table 3 shows they were chosen as the best model only five times. 1-step forecasting of Count of unique IPs uses the log diff scaling method, and 1-step, 2-steps, 5-steps, and 10-steps forecasting of Port 443 uses the mean std scaling method. We can see that Encoder-Decoder network types were performing overall better in more cases.

	dataset scaling	count
1-step	log diff	10
	log	3
	mean std	2
2-steps	log diff	5
	log	3
	mean std	7
5-steps	log diff	3
	log	4
	mean std	6
10-steps	log diff	3
	log	5
	mean std	7

Table 4. Comparison of counts of best scaling methods per all time series.

Table 4 shows the best performing data scaling methods in every 1-step, 2-steps, 5-steps, and 10-steps forecasting. It shows how many times did scaling method performs the best result per every time series. For example, for 1-step forecasting log diff scaling methods have the best results ten times, log scaling methods three times and mean std scaling methods two times. Results can also be seen in Table 5.

Although we work with the best combination of neural network types and data scaling methods for time series in the research, we cannot conclude which neural network type and data scaling methods are better. The combinations of these methods had comparable forecasting accuracy and the majority of time series are unpredictable (more details in the following sections).

We should note that a slight modification of the denominator in MASE was made to make the forecasting of neural networks and statistical methods comparable. Because of the difference between the size of the training dataset in the case of statistical

	Count of all alerts	COU IPs	COU IPs (AP)	recon scanning	attempt login
stat models	E - 0.6904	E - 0.7336	E - 0.6958	Ew - 0.8058	A - **1.0462**
log diff	Conv1DS - **0.6534**	DN - **0.7054**	GRU - **0.6505**	e1d1 - **0.7690**	GRU - 1.0550
log	e1d1 - 0.6553	Conv1D - 0.7191	Conv1D - 0.6619	e2d2 - 0.7758	LSTM - 1.1027
mean std	LSTM - 0.6605	GRU - 0.7230	e1d1LN - 0.6637	e1d1 - 0.7823	e1d1 - 1.0678
	attempt exploit	Port 22	Port 23	Port 80	Port 445
stat models	Ew - 0.6421	N - 1.6024	AEw - 0.7286	AEw - 1.0964	AE - 0.5920
log diff	e2d2 - 0.6429	edConv1D - **1.5988**	e1d1LN - 0.6964	LSTM - **1.0790**	e2d2 - **0.4209**
log	e1d1LN - **0.6246**	GRU - 1.6950	Conv1DS - 0.6941	LSTM - 1.0944	Conv1DS - 0.4997
mean std	GRU - 0.6260	edConv1D - 1.6476	e1d1LN - **0.6898**	e1d1LN - 1.0974	DN - 0.6418
	TCP	SSH	ICMP	MS WBT Server	telnet
stat models	E - 0.9661	Aw - **1.1321**	Aw - 0.6323	Ew - 1.0141	AE - 0.7789
log diff	e1d1 - **0.8316**	e1d1 - 1.1336	e2d2 - 0.6491	LSTM - 1.0058	e2d2 - 0.7788
log	LSTM - 0.8772	e1d1LN - 1.2428	GRU - **0.6178**	LSTM - 0.9993	e1d1LN - **0.7490**
mean std	e1d1LN - 0.8636	edConv1D - 1.3046	GRU - 0.6179	e1d1 - **0.9847**	e1d1 - 0.7540

Table 5. MASE comparison between best statistical method and best neural network models for 1-step prediction for all time series. Notes: COU - Count of unique; AP - another approach; A - ARIMA model; E - Exponential Smoothing; N - naive model; AE - ARIMA + Exponential smoothing (average); w - rolling window.

	Count of all alerts	COU IPs	COU IPs (AP)	recon scanning	attempt login
stat models	AEw - 0.9578	Aw - 1.1651	AEw - 0.9492	AEw - 1.0784	A - 3.3308
log diff	GRU - 0.9408	GRU - 1.1880	GRU - 0.8771	GRU - 1.0629	e2d2 - 3.4062
log	e1d1 - 0.8854	e1d1 - 1.0903	Conv1D - 0.8578	GRU - **0.9483**	Conv1DS - **3.3129**
mean std	e1d1 - **0.8169**	e1d1 - **1.0371**	e1d1 - **0.8358**	e2d2 - 0.9760	edConv1D - 3.8772
	attempt exploit	Port 22	Port 23	Port 80	Port 445
stat models	Ew - 0.6453	N - 3.3959	AEw - 1.1520	E - 1.8115	Ew - 0.6520
log diff	e1d1 - 0.6507	edConv1D - **3.3749**	e2d2 - 1.2185	Conv1DS - 1.7963	LSTM - **0.6025**
log	e2d2 - 0.6242	Conv1DS - 3.3820	e1d1 - **1.0883**	e1d1 - 1.8320	Conv1DS - 0.6065
mean std	e1d1 - **0.6210**	edConv1D - 3.7218	e2d2 - 1.0983	Conv1DS - **1.7567**	DN - 0.7080
	TCP	SSH	ICMP	MS WBT Server	telnet
stat models	A - 1.1029	Aw - **3.6500**	Aw - 0.6300	AE - 1.8209	AE - 0.8734
log diff	e1d1LN - 1.0453	Conv1D - 3.6610	e2d2 - 0.6770	e1d1LN - 1.8096	e1d1 - 0.9549
log	e1d1 - **1.0144**	Conv1DS - 3.9079	e2d2 - 0.6278	e2d2 - 1.8269	e2d2 - **0.8460**
mean std	e1d1LN - 1.0584	edConv1D - 4.1247	e2d2 - **0.6267**	Conv1DS - **1.7559**	edConv1D - 0.8750

Table 6. MASE comparison between best statistical method and best neural network models for 10-steps predictions for all time series. Notes: COU - Count of unique; AP - another approach; A - ARIMA model; E - Exponential Smoothing; N - naive model; AE - ARIMA + Exponential smoothing (average); w - rolling window.

methods and neural network models, the calculations of the denominator in equation 2 were based on 1200 values used for fitting statistical models.

Results of the third stage of the evaluation

At this stage, we use the results from the previous phase (MASE criterion and visual representations) to divide 15-time series into three categories:

- well predictable time series,
- unpredictable time series, and
- unpredictable noise.

The first group of time series consists of well predictable time series in terms of future security events. This group of time series has only one representative - 445 port. For this group, the MASE criterion is the best of all examined time series. In the 1-step forecasting, neural network-based methods approached to value 0.4209 and statistical forecasting models approached to value 0.5920. For 2-steps, 5-steps, and 10-steps forecasting, the values are less or equal than 0.6025 for neural network-based methods and 0.6521 for statistical forecasting methods.

An illustration of the 1-step forecasting for this group of time series is shown in Figure 1. This Figure shows 1-step forecasting based on e2d2 network with log diff scaling and combination of ARIMA and Exponential smoothing. On the other hand, an illustration of the 10-steps forecasting for this group of time series is shown in Figure 2.

Port 445/TCP is connected to the SMB (Server Message Block) protocol, a communication protocol for providing shared access to files, printers, and serial ports between nodes on a network. This port is used by several types of malware (e.g. trojan horses, ransomware) for their spreading. WannaCry malware or Emotet botnet used this port within their malicious activities and spreading. It is an interesting finding that time series characterising malware or botnet spreading is well predictable.

The second group of time series represents the time series, in which the forecasting methods are comparable to the naive forecasting with drift. This group of time series consists of 12 of the 15 examined time series (e.g. Count of all alerts, Count of IP addresses, Recon scanning category, Attempt login category, Port 22, Port 23, and Port 80).

According to MASE criterion, this group of time series can be divided into two subgroups. The first subgroup is characterized by MASE values in 1-step forecasting from 0.6904 to 0.8058 (statistical methods) and from 0.6553 to 0.7690 (methods based on neural networks). However, the graphical

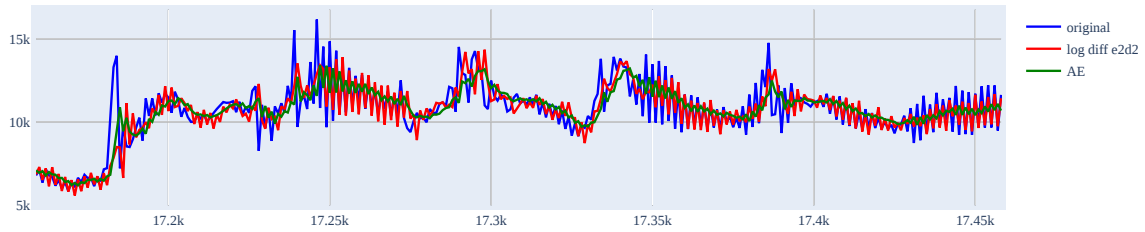


Fig. 1. 1-step forecasting for Port 445 time series based on e2d2 network with log diff scaling and combination of ARIMA and Exponential smoothing.

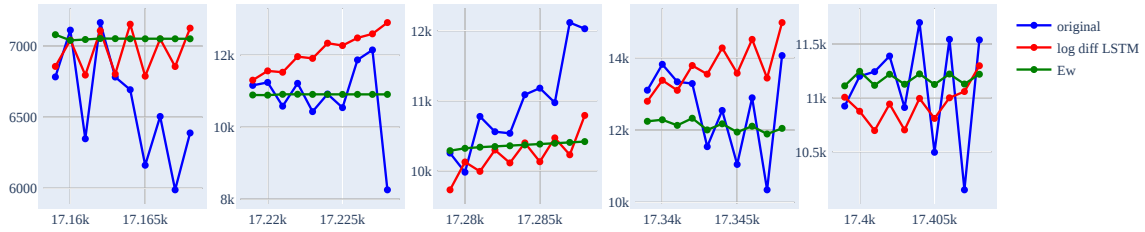


Fig. 2. 10-steps forecasting for Port 445 time series based on LSTM network with log diff scaling and combination of Exponential smoothing with rolling window.

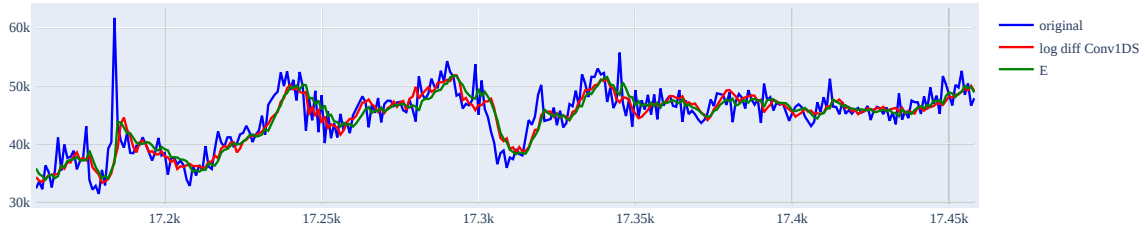


Fig. 3. 1-step forecasting for Count of all alerts time series based on Conv1DS network with log diff scaling and Exponential smoothing.

representation (Figure 3) indicates that the given methods are comparable to naive forecasting with drift. In 10-steps forecasting (Figure 4), the values are around one or slightly above it.

The second subgroup is characterised by the fact that MASE values are above 1 in 1-step forecasting and gradually increase to 10-steps forecasting. Those values increase. An illustration of the 1-step forecasting for this group of time series is shown in Figure 5 and the 10-steps forecasting for this group of time series is shown in Figure 6.

Time series in this group can be characterised as ports (22/TCP, 23/TCP, 80/TCP) or categories (Recon scanning, Attempt login) commonly used by attackers for reconnaissance (MITRE ATT&CK tactic - TA0043) initial access (MITRE ATT&CK tactic - TA0001). Since forecasting methods do not give valuable results in this group of time series, they can be considered unpredictable.

The last group of time series represents the time series, which can be characterised as unpredictable noise. This group of time series consists of: Attempt exploit category and ICMP protocol time series.

In this group of time series (as shown in Figure 7 and Figure 8), it is not possible to forecast any future values. Although these time series have one the lowest MASE values (about 0.6200 for neural network forecasting methods and 0.6400 for statistical forecasting methods) except for the well predictable time series group, it is impossible to predict any value. Good MASE values are due to the low difference between the actual and predicted value. It indicates that, in addition to the MASE value, it is necessary to consider the MAE criterion as it provides additional information on the quality of forecasting. Moreover, it is possible to apply appropriate statistical tests (e.g. Box-Ljung, Box-Pierce) to verify if there are some (linear) dependencies in considered time series or not. If the correlations are zero and the time-series variance is stable, it can be regarded as white noise.

An illustration of the 1-step forecasting for this group of time series is shown in Figure 7. This Figure shows 1-step forecasting of attempt exploit based on e1d1LN with log scaling and Exponential smoothing with rolling window. On the other hand, an illustration of the 10-steps forecasting for this group of time series is shown in Figure 8.

Conclusion and future works

In our research, we have focused on using time series in NSSA forecasting based on statistical methods and neural networks methods. All the results presented in this paper are related to a specific area of cyber security - NSSA based on data collected and analysed by security data-sharing platforms.

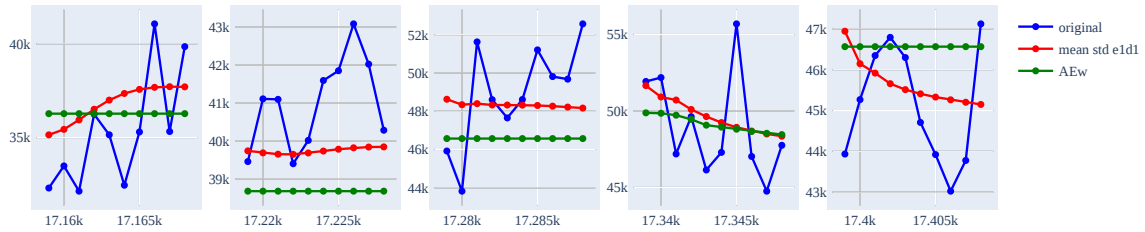


Fig. 4. Ten-step forecasting for Count of all alerts time series based on e1d1 network with mean std scaling and combination of ARIMA and Exponential smoothing with rolling window.

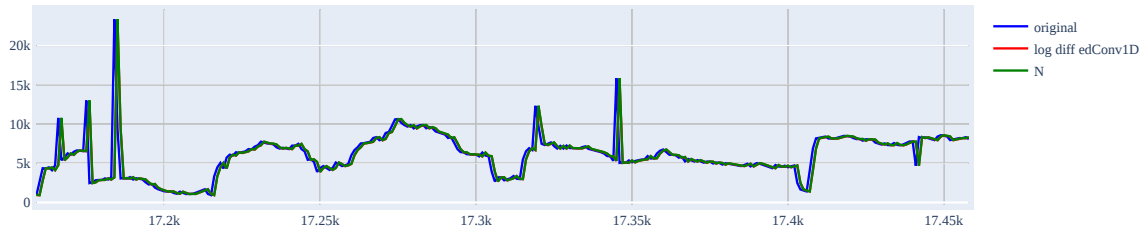


Fig. 5. 1-step forecasting for Port 22 time serie based on edConv1D network with log diff scaling and Naive approach.

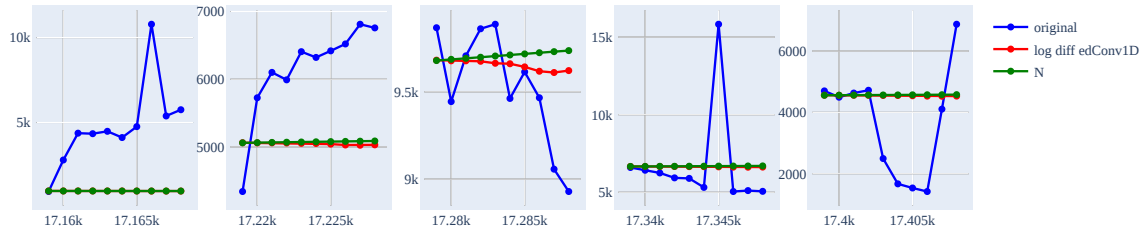


Fig. 6. 10-steps forecasting for Port 22 time serie based on edConv1D network with log diff scaling and combination of Naive approach.

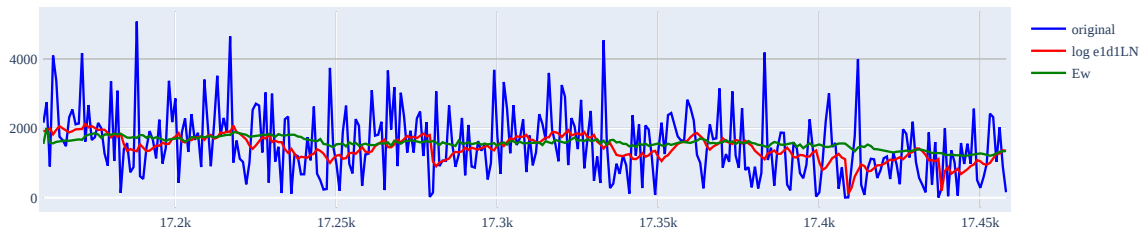


Fig. 7. 1-step forecasting for attempt exploit time serie based on e1d1LN network with log scaling and Exponential smoothing with rolling window.

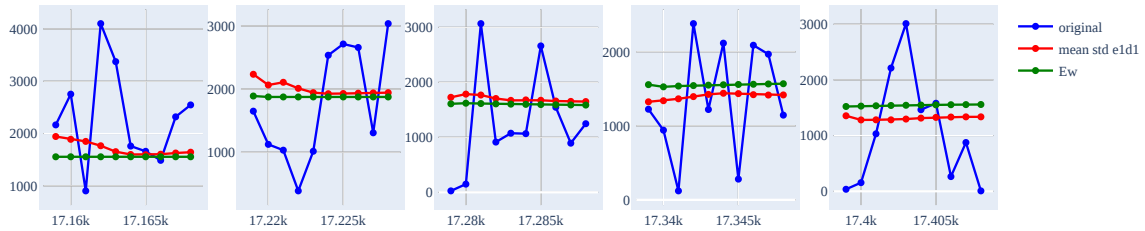


Fig. 8. 10-steps forecasting for attempt exploit time series based on e1d1 network with mean std scaling and combination of Exponential smoothing with rolling window.

In this paper, we compared tree data scaling methods, but we can not conclude which one is best because of the unpredictability of our time series. On the other hand, our experiments showed that Encoder-Decoder neural networks are better than standardly used recurrent networks (GRU, LSTM) for NSSA time series forecasting, but it is questionable because of the unpredictability of the time series we used.

As one result of this paper, we have divided individual time series into three categories according to the possibility to forecast future values. We have shown that it is necessary to proceed to a more detailed analysis of time series in terms of specific criteria (e.g. MASE) and a graphical representation of the forecasting. Time series of security alerts linked to ports (22/TCP, 23/TCP, 80/TCP) or categories (recon scanning, attempt login) commonly used by attackers for initial reconnaissance access can be considered unpredictable. Another example is time series that can be regarded as unexpected noise (e.g. ICMP reconnaissance attacks, exploitation attempt category).

Our future research will concern the evaluation of the findings from this paper for real-time NSSA forecasting. We want to extend our research with another type of real security data to generalise our results. Also, we would like to use other types of methods for the time series forecasting (e.g. support vector machines, Bayesian networks, neural network models Transformers) and specify more complex criteria for time series creation.

Competing interests

There is NO Competing Interest.

Author contributions statement

P.P. collected and processed data, R.S. and A.G. conducted the experiment(s), P.S., R.S. and A.G. analysed the results. P.S. and R.S. wrote and reviewed the manuscript.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions. This work is supported in part by funds from the VVGS projects under contract No. VVGS-PF-2020-1423 and No. VVGS-PF-2020-1427 and Slovak Research and development agency project under contract No. APVV-17-0561.

References

1. Mohamed Abdlhamed, Kashif Kifayat, Qi Shi, and William Hurst. Intrusion prediction systems. In *Information Fusion for Cyber-Security Analytics*, pages 155–174. Springer, 2017.
2. Tim Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99–105, 2000.
3. Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.
4. George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
5. Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2016.
6. Jason Brownlee. *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.
7. Kai Chen, Yi Zhou, and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. In *2015 IEEE international conference on big data (big data)*, pages 2823–2824. IEEE, 2015.
8. Jin-Hee Cho, Dilli P Sharma, Hooman Alavizadeh, Seunghyun Yoon, Noam Ben-Asher, Terrence J Moore, Dong Seong Kim, Hyuk Lim, and Frederica F Nelson. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Commun. Surv. Tutor*, 22(1):709–745, 2020.
9. Vasiliki Christou and Konstantinos Fokianos. On count time series prediction. *Journal of Statistical Computation and Simulation*, 85(2):357–373, 2015.
10. Edward Condon, Angela He, and Michel Cukier. Analysis of computer security incident data using time series models. In *Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on*, pages 77–86. IEEE, 2008.
11. J Do Croston. Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303, 1972.
12. Francis X Diebold and Robert S Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002.
13. K Drachal. multmdm: Multivariate version of the diebold-mariano test. 2018.
14. Mica R Endsley. Situation awareness global assessment technique (sagat). In *Proceedings of the IEEE 1988 national aerospace and electronics conference*, pages 789–795. IEEE, 1988.
15. Xing Fang, Maochao Xu, Shouhuai Xu, and Peng Zhao. A deep learning framework for predicting cyber attacks rates. *EURASIP Journal on Information Security*, 2019(1):1–11, 2019.
16. Wei Feng, Yuqin Wu, and Yexian Fan. A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit. *International Journal of Intelligent Computing and Cybernetics*, 2018.
17. Pavel Filonov, Fedor Kitashov, and Andrey Lavrentyev. Rnn-based early cyber-attack detection for the tennessee eastman process. *arXiv preprint arXiv:1709.02232*, 2017.
18. Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. *arXiv preprint arXiv:1612.06676*, 2016.
19. Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE, 2016.
20. Palash Goyal, KSM Hossain, Ashok Deb, Nazgol Tavabi, Nathan Bartley, Andr'es Abeliuk, Emilio Ferrara, and Kristina Lerman. Discovering signals from web sources to predict cyber attacks. *arXiv preprint arXiv:1806.03342*, 2018.
21. Fannv He, Yuqing Zhang, Donghang Liu, Ying Dong, Caiyun Liu, and Chensi Wu. Mixed wavelet-based neural network model for cyber security situation prediction using modwt and hurst exponent analysis. In *International Conference on Network and System Security*, pages 99–111. Springer, 2017.

22. Martin Husák, Václav Bartoš, Pavol Sokol, and Andrej Gajdoš. Predictive methods in cyber defense: Current experience and research challenges. *Future Generation Computer Systems*, 115:517–530, 2021.
23. Martin Husák, Jana Komárková, Elias Bou-Harb, and Pavel Čeleda. Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys & Tutorials*, 21(1):640–660, 2018.
24. Martin Husák, Martin Žádník, Václav Bartoš, and Pavol Sokol. Dataset of intrusion detection alerts from a sharing platform. *Data in Brief*, 33:106530, 2020.
25. Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.
26. Rob J. Hyndman, Yeasmin Khandakar, et al. *Automatic time series for forecasting: the forecast package for R*. Number 6. Monash University, Department of Econometrics and Business Statistics, 2007.
27. Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.
28. Pavel Kacha. Idea: security event taxonomy mapping. In *18th International Conference on Circuits, Systems, Communications and Computers*, 2014.
29. Pavel Kacha, Michal Kostenec, and Andrea Kropacova. Warden 3: Security event exchange redesign. In *19th International Conference on Computers: Recent Advances in Computer Science*, 2015.
30. Taewook Kim and Ha Young Kim. Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. *PloS one*, 14(2):e0212320, 2019.
31. Ji-Bao Lai, Hui-Qiang Wang, Xiao-Wu Liu, Ying Liang, Rui-Juan Zheng, and Guo-Sheng Zhao. Wnn-based network security situation quantitative prediction method and its optimization. *Journal of computer science and technology*, 23(2):222–230, 2008.
32. Daria Lavrova, Dmitry Zegzhda, and Anastasiia Yarmak. Using gru neural network for cyber-attack detection in automated process control systems. In *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–3. IEEE, 2019.
33. Yu-Beng Leau and Selvakumar Manickam. Network security situation prediction: a review and discussion. In *International Conference on Soft Computing, Intelligence Systems, and Information Technology*, pages 424–435. Springer, 2015.
34. Zongming Lin, Guolong Chen, Wenzhong Guo, and Yanhua Liu. Pso-bpnn-based prediction of network security situation. In *2008 3rd International Conference on Innovative Computing Information and Control*, pages 37–37. IEEE, 2008.
35. Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018.
36. Kishan G Mehrotra, Chilukuri K Mohan, and HuaMing Huang. *Anomaly Detection Principles and Algorithms*. Terrorism, Security, and Computation. Springer, 2017.
37. Navruzov, Fedor and Halytskyi, Vladyslav. Seq2seq models for time-series forecasting with tensorflow. `https://docs.google.com/presentation/d/1EK4MYilx8RfwRCfbPPolMYBnS8oQvBF9DdbOBOThJVs`, 2019. slide 12.
38. Ahmet Okutan, Gordon Werner, Katie McConky, and Shanchieh Jay Yang. Poster: Cyber attack prediction of threats from unconventional resources (capture). In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2563–2565, 2017.
39. Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, and Victor Chang. An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, 76(3):2098–2118, 2020.
40. Patrik Pekarčík, Andrej Gajdoš, and Pavol Sokol. Forecasting security alerts based on time series. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 546–557. Springer, 2020.
41. Thulasy Ramiah Pillai, Sellappan Palaniappan, Azween Abdullah, and Hafiz Muhammad Imran. Predictive modeling for intrusions in communication systems using garma and arma models. In *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, pages 1–6. IEEE, 2015.
42. Yingying Qi, Wenli Shang, and Xiaojun He. A combined prediction method of industrial internet security situation based on time series. In *Proceedings of the 2019 the 9th International Conference on Communication and Network Security*, pages 84–91, 2019.
43. Pavol Sokol and Andrej Gajdoš. Prediction of attacks against honeynet based on time series modeling. In *Proceedings of the Computational Methods in Systems and Software*, pages 360–371. Springer, 2017.
44. Pavol Sokol, Lenka Kleinová, and Martin Husák. Study of attack using honeypots and honeynets lessons learned from time-oriented visualization. In *IEEE International Conference on Computer as a Tool (EUROCON)*, pages 1–6. IEEE, 2015.
45. Chenghua Tang, Xin Wang, Reixia Zhang, and Yi Xie. Modeling and analysis of network security situation prediction based on covariance likelihood neural. In *International Conference on Intelligent Computing*, pages 71–78. Springer, 2011.
46. Gordon Werner, Ahmet Okutan, Shanchieh Yang, and Katie McConky. Forecasting cyberattacks as time series with different aggregation granularity. In *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7. IEEE, 2018.
47. Gordon Werner, Shanchieh Yang, and Katie McConky. Time series forecasting of cyber attack intensity. In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, page 18. ACM, 2017.
48. Gordon Werner, Shanchieh Yang, and Katie McConky. Leveraging intra-day temporal variations to predict daily cyberattack activity. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 58–63. IEEE, 2018.
49. Maochao Xu, Kristin M Schweitzer, Raymond M Bateman, and Shouhuai Xu. Modeling and predicting cyber hacking breaches. *IEEE Transactions on Information Forensics and Security*, 13(11):2856–2871, 2018.
50. Shanchieh Jay Yang, Haitao Du, Jared Holsopple, and Moises Sudit. Attack projection. In *Cyber Defense and Situational Awareness*, pages 239–261. Springer, 2014.
51. Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security*, 10(8):1666–1677, 2015.

52. Haibo Zhang, Qing Huang, Fangwei Li, and Jiang Zhu. A network security situation prediction model based on wavelet neural network with optimized parameters. *Digital Communications and Networks*, 2(3):139–144, 2016.
53. Yaxing Zhang, Shuyuan Jin, Xiang Cui, Xi Yin, and Yi Pang. Network security situation prediction based on bp and rbf neural network. In *International Conference on Trustworthy Computing and Services*, pages 659–665. Springer, 2012.
54. Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.
55. Ruijuan Zheng, Dan Zhang, Qingtao Wu, Mingchuan Zhang, and Chunlei Yang. A strategy of network security situation autonomic awareness. In *International Conference on Network Computing and Information Security*, pages 632–639. Springer, 2012.

Pavol Sokol. He is an assistant professor at Pavol Jozef Šafárik University in Košice, head of CSIRT-UPJS (Computer Security Incident Response Team of Pavol Jozef Šafárik University in Košice) and a member of The Honeynet Project. He received a PhD degree in Computer Science from the same university. His research interests are related to cybersecurity (honeypots and cyber situational awareness) and the legal aspects of information and communications technologies (particularly on the protection of privacy and liability in cyberspace).

Richard Staňa. He is Ph.D. student at Pavol Jozef Šafárik University in Košice, head of Computer network administration ŠDaJ UPJŠ and network administrator at CAI UPJŠ. His PhD. thesis is mainly about using machine learning for time series forecasting in the field of cybersecurity.

Andrej Gajdoš. He is an assistant professor at Pavol Jozef Šafárik University in Košice, Institute of Mathematics, where he received a PhD in Applied Mathematics. His research interests are probability and mathematical statistics. He focuses on modelling and prediction of time series using kriging. He deals with data analysis within cybersecurity using various time series, cluster analysis, and machine learning methods.

Patrik Pekarčík. He is a PhD student at the Institute of Computer Science and holds a master's degree in Computer science from the same faculty. In his previous studies, he engaged in research into software development with a focus on automation tools (bachelor thesis). Later, he developed a brand-new way of programming Arduino devices called component-oriented and Event-driven Arduino programming (master's thesis). Currently, in PhD studies, he is working on real-time security data processing. He assists in the course Software projects, where he also leads several active student development projects.

Network security situation awareness forecasting based on neural networks

Richard Staňa[1], Patrik Pekarčík[2], Andrej Gajdoš[3], and Pavol Sokol[4]

[1] Pavol Jozef Šafárik University in Košice, Faculty of Science, Košice, Slovakia
richard.stana@upjs.sk
[2] Pavol Jozef Šafárik University in Košice, Faculty of Science, Košice, Slovakia
patrik.pekarcik@upjs.sk
[3] Pavol Jozef Šafárik University in Košice, Faculty of Science, Košice, Slovakia
andrej.gajdos@upjs.sk
[4] Pavol Jozef Šafárik University in Košice, Faculty of Science, Košice, Slovakia
pavol.sokol@upjs.sk

Abstract. The increasing number of cybersecurity threats affects the security situation of organisations. The maintenance of the operational picture of the organisation, which integrates all relevant information for selecting appropriate countermeasures, becomes a vital role for organisations. In this paper, we focus on network security situation awareness forecasting. The paper aims to answer two questions - the influence of loss function in neural networks on network security situation awareness forecasting and a comparison of statistical methods and neural networks in network security situation awareness forecasting. For this purpose, we used two-time series representing cybersecurity alerts collected by system Warden. This paper shows an analysis according to which the MAE and MASE loss functions give better results than MSE. Also, we can state that neural networks are more accurate for network security situation awareness forecasting.

Keywords: Cybersecurity, Network security, Network security situation awareness, Forecasting, Time series

1 Introduction

Nowadays, the number of new cybersecurity threats and cybersecurity incidents is on the rise. The main goal of organisations' security teams is to prevent cybersecurity incidents or minimise their impact. For example, the organisations' network administrators or security teams may prevent these incidents by disallowing the specific network protocols or updating systems to address security vulnerabilities. In this respect, we observe a trend of transition from reactive activities to proactive activities [1].

An important element in ensuring the proactive activities of the organisation is the maintenance of the operational picture of the organisation, which integrates all relevant information for identifying attacks and selecting appropriate

countermeasures [2]. This operational picture can be defined as network security situation awareness (NSSA). Bass et al. introduced the origin, concept, target and characteristics of NSSA in more detail in [3].

According to a different perception of an object, NSSA can be divided into the network security situation assessment and network security situation forecasting [4]. Forecasting the security situation is an essential part of the NSSA and allows anticipating cybersecurity attacks and cybersecurity threats. It provides network administrators and security teams time to make adequate decisions on their next steps. Overall, this allows better analysing security threats and management of cybersecurity incidents.

Researchers have proposed and used various approaches to forecast network security situation awareness in recent years, such as statistical methods, game theory methods, or neural networks. In the following section, we focus on state of the art in statistical methods and neural networks in more detail. At the same time, there are some problems in these methods, such as the loss of network data information caused by situation assessment and the low forecasting accuracy of the neural network model used for the NSSA forecasting [5]. To improve the accuracy of the NSSA forecasting, this paper aims to: (I) analyse the influence of loss function in neural networks on the NSSA forecasting and (II) compare statistical methods and neural networks in NSSA forecasting.

This paper is based on previous research [6,7]. Within this paper, we assume the fact that in the NSSA forecasting, there is a lot of time series forecasting with neural networks that look like naive forecasting with drift [8]. Definition of the mean absolute scaled error (MASE) shows that it compares forecasting with naive forecasting. Using MASE as a loss function, we can "punish" neural network when its forecasting looks like naive forecasting with drift.

This paper is organized into six sections. Section 2 reviews state of the art in network security awareness forecasting. Section 3 is devoted to research methodology and outlines the dataset and methods used for the analysis. Section 4 states the experimental evaluation. Section 5 discusses the results. The last section concludes the paper and discusses the challenges for future research.

2 Related works

This section overviews papers and research groups' activities related to network security situation awareness forecasting. This section is divided into two parts: the statistical time series approach and the neural networks approach. Most of the papers focus on the detection of attacks rather than a prediction of attacks or NSSA forecasting [9].

In the field of the NSSA, the Auto-Regressive Integrated Moving Average (ARIMA) models are a very frequently used approach. Examples of research work using these forecasting methods are [10,11,12]. Above mentioned ARIMA models are often used in combination with other models. For example, ARIMA models are used with the Bayesian Networks to predict future cyber attack (malware, malicious URL, and malicious e-mail) occurrences [13]. Another example

is a combination of ARIMA models and gray-box models. In the paper [14], the authors responded to the disadvantages of the separate usage of these models. ARIMA models require strict inputs, and the gray-box models do not consider the system's randomness. This combination is used in the extreme-value phenomenon analysis [15]. An exciting combination of methods for forecasting purposes is used in several research papers. The analysis of the fitting of ARMA and GARMA models to the cyber-attack process is an objective of paper [16].

Neural networks are commonly used in the field of time series prediction in cybersecurity. There are a lot of papers that use older types of smaller feed-forward networks or wavelet neural networks trained by backpropagation or genetics algorithm (and its variants) to forecast network security situations (e.g., [17,18]). On the other hand, modern approaches like recurrent neural networks (GRU, LSTM) were used in the paper [19] for forecasting the network security situation. In the paper [20], authors compare the ARIMA approach, LSTM, and GRU neural networks for cyberattack prediction. This prediction is based on the combination of time series and external signals. Another research paper [21] predicts time series based on data collected by the honeypot. For this prediction, the authors used a bi-directional LSTM neural network. Several research groups have been working with recurrent neural networks like LSTM and GRU to predict cyberattacks based on time series created from industrial data [22,23,24].

3 Methodology

3.1 Dataset

Our research used a dataset collected and preprocessed by a Warden system [25]. This system was created for sharing cybersecurity alerts between hosts connected to this sharing system. Security alerts are stored in a descriptive data model using a key-value JSON extensible structure called IDEA (Intrusion Detection Extensible Alert) format [26]. Primary data sources for the Warden system may include honeypots, intrusion detection systems, network flow probes, system log records, and other sensors and data sources. The data used in the research are collected from real operation in the computer networks of the Czech national research and education network and other Czech commercial organizations.

Security alerts in the IDEA format contain several mandatory fields (form, ID, detect time, category) [26] and many optional fields. The fields we used in this experiment are the category of security alert, source and destination IP addresses, source and destination ports, network protocol, and detection time. The Warden system collected the data we used in this research for one year (from 2017-12-11 to 2018-12-11). Our dataset contains approximately one billion security alerts from various data sources (mainly honeypots).

In our research, we used time series with 30 minutes time period. We deal with the creation of time series and selection of periods in more detail in the papers [27,6]. Also, we used two selected time series, such as time series representing the total number of alerts and time series representing alerts related to

the services running on port 445/TCP. These time series are representatives of two categories of time series for the area of NSSA forecasting (well predictable time series, and unpredictable time series) [8].

3.2 Method Description

There is a wide range of quantitative forecasting methods, and their usage often depends on the specific disciplines, the nature of data, or specific purposes. Our research compared the accuracy of three different loss functions mean absolute error (MAE), mean squared error (MSE) and MASE, by implementing five different neural networks to obtain predictions. After that, we compare the best methods with usually used statistical methods for time series forecasting. From neural networks, we employ five types of neural networks: dense network; LSTM; GRU; Convolutional neural networks; Encoder-Decoder networks. From the statistical method we choose: ARIMA models; Exponential smoothing models (state-space models); the naive approach (with drift); Combination (average) of ARIMA and Exponential smoothing models. A complete description of the mentioned architectures can be found below.

Neural networks

There is a lot of work done in time series forecasting with neural networks. For example in field of stock prediction [28,29,30], traffic prediction [31,32], etc. We developed five multi-layers neural networks, most of them were inspired by [33] and similar networks were previously used in our work [8].

In the following text, we provide a description of the architectures (abbreviations, used later and denoting individual architectures are in parentheses):

- Dense network (DN) - 4 dense layers (1024, 512, 256, 128 units, activation relu), 1 dense layer (1 unit, activation linear);
- Long Short-Term Memory (LSTM) - 3 LSTM layers (256, 256, 256 units, default parameters), 1 dense layer (1 unit, activation linear);
- Gated Recurrent Unit (GRU) - 3 GRU layers (256, 256, 256 units), 1 SimpleRNN layer (128 units), 1 dense layer (1 unit, activation linear);
- 1D convolution (Conv1D) - 3 Conv1D layers (256, 256, 256 filters, 3, 3, 3 kernel size, activation relu, pading same), 1 dense layer (64 unit, activation relu), 1 dense layer (1 units, activation linear);
- Encoder-Decoder LSTM (e1d1) - 1 LSTM encoder layer (512 units encoder, return state True), RepeatVector layer, 1 LSTM decoder layer (512 units encoder, return state True), TimeDistributed (1 dense unit, activation linear)

Statistical methods

The choice of statistical methods for this research is based on our previous research activity [27,8]. ARIMA and Exponential Smoothing (ETS) [34] are the

most commonly used statistical models in the modeling and time series prediction classes.

ARIMA models represent a generalization of the ARMA model class, including a wide range of non-stationary series. These models ensure the stationarity of the time series by a finite number of differentiations. ARMA models are a combination of automatic regression (AR) and moving average (MA) [35].

The ETS class provides additional access to time series modeling and forecasting. Prediction using models in this class is characterized by a weighted combination of older observations with new ones. The new observations have a relatively higher weight compared to the older observations. Exponential smoothing reflects that weights decrease exponentially with the age of the observations. On the one hand, ETS models are based on trend descriptions and seasonality in the data. On the other hand, ARIMA models aim to describe autocorrelations in the book [34] data.

In the research, we also use the naive methods [36,35] as a benchmark for statistical methods. These methods can process large datasets and, at the same time, do not have high computational demands. We also added a combination (average) of ARIMA and ETS methods to the experiments to compare standard methods with their diversity. The idea of averaging or increasing is currently nothing new [**?**].

4 Experiment evaluation

We consider only one step ahead predictions.

For forecast accuracy evaluation, we employ two commonly used metrics - MASE used [37] and MAE.

MASE is a preferred metric as it is less sensitive to outliers, more easily interpreted and less variable on small samples. MASE is defined as [34]:

$$\text{MASE} = \text{mean}(|q_j|) \tag{1}$$

where q_j is:

$$q_j = \frac{e_j}{\frac{1}{T-1}\sum_{i=2}^{T}|y_i - y_{i-1}|}, \tag{2}$$

where y_i represents observed value, T is the length of time series.

For a better view of accuracy in both time series, we take into account also MAE, which is defined as follows [34]:

$$\text{MAE} = \text{mean}(|e_j|) \tag{3}$$

In both cases, e_j is forecast error, i.e., the difference between an observed value and its forecast.

Both time series we used consist of 17473 values. We did not use the first 27 and last 14 values because there were primarily zeros or missing values. Due to missing values between 15550 and 15601 in the whole dataset, we split the

dataset into three parts. In the first part, there were values between 28 and 15549 (15522 values), the second part included values between 15602 and 16601 (1000 values), and the last part contained values between 16602 and 17458 (857 values).

The first and the second part was used for training neural networks. The third part was used for testing. During neural networks training, we employed a window containing 384 values (8 days) for every model and time series. We trained all five neural networks in 40 epochs. From every type of network, we trained four instances with Adam optimiser, two with fixed learning rate (lr) to 0.001 and two with decreasing lr from 0.001 to 0.0001 decreasing by two when testing loss did not decrease in four epochs. If testing loss of a particular neural network had a decreasing tendency (at the end of training), we trained it for more than 20 epochs. After training, we choose the best model from four instances based on MASE metrics.

It is essential to prepare dataset before training a neural network. To our time series, we applied Standardisation (subtraction of mean and division by standard deviation - mean and standard deviation were calculated using the first part of the dataset). For neural networks training, we employed three different loss functions, MAE, MSE and MASE. For both, we used standard implementation, which in TensorFlow pages. In our implementation of the MASE loss function, we first describe the predicted value and real value as inputs. This was done because we wanted to calculate MASE according to unscaled data. Then MASE was implemented as described by equations 1 and 2.

GPU NVidia GTX 1080 and 1060, Keras and TensorFlow [38] version 2.4 were used to train neural networks. The batch size was set to 128. To make models comparisons easier, we used the tool Weights & Biases [39]. In total, we trained more than 120 neural networks (5 networks described in Methodology x 3 loss functions x 4 instances x 2-time series).

Additionally, we compared predictions based on neural networks with forecastings based on statistical approaches described in the previous section. Due to missing data in the dataset, long training time when using the extensive dataset and weak impact of older data on statistical methods, we used only values from the second part of the dataset for fitting statistical methods (as described in the previous section). We used values from the third part of the dataset to test their forecasting accuracy. On the other side, we train neural networks on both (first and second) parts of the dataset because, generally, more data means better results from neural networks. With more data, neural networks can find more patterns in data, generalize them better and get better results, even with older data.

The methods were evaluated according to principles and implementations presented in our previous work [6,27,8]. For our research, we used R functions from one of the most common R-packages for time-series predictions called *forecast* [40]. This package contains valuable features when working with large data sets or potentially in real-time prediction. In addition, these functions are used to adjust ARIMA and ETS model classes automatically. These functions are

designed to automatically select the best model from the considered class under the given conditions, for example, considering the information criterion [40].

In the next part of our research, we focused on two ways of adapting statistical models: the classical model and the "rolling window" approach. With the classic model, we kept the entire training data sets. Step by step, we added one more observation to the training set in each round of evaluation. In the second method, we focused on the "rolling window" approach ("one in, one out" approach). As in the previous method, we added one new observation from the test set to the training set. The difference was that in each round of evaluation, we removed the oldest observation from the training set.

Seasonality was not taken into account due to its minimal impact on forecasting performance as shown in paper [27] where we used the same dataset.

At this place, it is essential to note that we have modified the denominator in MASE. The reason was the difference between the size of the training data set in the case of statistical methods and neural network models. The aim was to achieve comparability of forecasting for both approaches. For this reason, the denominator calculations in the equation 2 were based on the 1000 training values used to adjust the statistical models.

5 Results and Discussion

In this section, we compared the results which were obtained according to the description in the previous section. Because MAE was used as a metric, we present some statistical information about the dataset:

- time series of the total number of alerts: minimum 22, maximum 155,818 and mean 34,594.25.
- time series of the alerts related to the services running on port 445/TCP: minimum 0, maximum 16,168 and mean 5,972.56.

test metrics	loss function	DN	LSTM	GRU	e1d1	Conv1D
MASE	MAE	**0.9950**	0.9213	**0.9286**	**0.9166**	**0.9254**
	MSE	1.0245	0.9442	0.9550	0.9430	0.9567
	MASE	1.0147	**0.9192**	0.9352	0.9178	0.9362
MAE	MAE	**2645.9203**	2449.9389	**2469.3886**	**2437.3081**	**2460.6580**
	MSE	2724.2048	2510.7505	2539.4539	2507.7080	2543.9238
	MASE	2698.3200	**2444.1826**	2486.8879	2440.6539	2489.3861

Table 1. MASE and MAE comparison for three loss functions on all neural networks forecasting for the total number of alerts on testing dataset. Every bold number is the best result for the actual neural network from 3 loss functions.

Tab. 1 and Tab. 2 show the results of comparison of neural networks models for selected time series (the total number of alerts - Tab. 1 and security

alerts related to the service running on network port 445/TCP - Tab. 2). In the analysis, we used MASE and MAE metrics to evaluate the results. Each neural network was used with a specific loss function. According to the results shown in the given tables, it can be stated that the MSE loss function shows the worst results in all investigated neural networks. The MAE loss function achieves the best results or approaches them. The MASE loss function implemented by us is comparable to the MAE loss function.

At the same time, we analysed statistical methods in the research. Their comparison according to MASE and MAE metric may be seen in Tab. 3. As may be seen from the results, the value of the MASE metric for NSSA forecasting in the time series of the number of cybersecurity alerts is bigger than 1. It means that the given forecasting method is worse than the average naive forecast. The time series of alerts associated with services running on port 445/TCP has another result. As may be seen from the table, the used models have a MASE metric value below 1. Exponential smoothing appears to be the best method in both cases. These results confirm the findings from previous research [6]. Similar time series were used in the current article, but with a different period.

test metrics	loss function	DN	LSTM	GRU	e1d1	Conv1D
MASE	MAE	**0.6972**	0.6633	0.6307	**0.6408**	**0.7080**
	MSE	0.7215	0.7118	0.7321	0.7038	0.8201
	MASE	0.7020	**0.6617**	**0.6210**	0.6582	0.7208
MAE	MAE	**1186.1808**	1128.4426	1072.9371	**1090.1418**	**1204.4519**
	MSE	1227.4236	1210.9351	1245.5377	1197.3586	1395.2064
	MASE	1200.1366	**1125.6894**	**1056.5102**	1119.8305	1226.3334

Table 2. MASE and MAE comparison for three loss functions on all neural networks forecasting port 445/TCP on the testing dataset. Every bold number is the best result for the actual neural network from three loss functions.

time series	the total number of alerts		port 445/TCP	
test metrics	MASE	MAE	MASE	MAE
A	1.0536	2801.7281	0.7950	1352.5694
Aw	1.0569	2810.3664	0.8046	1368.8168
E	**1.0319**	**2744.0770**	**0.7661**	**1303.3641**
Ew	1.0374	2758.7118	0.7741	1317.0408
AE	1.0411	2768.4691	**0.7661**	**1303.3641**
AEw	1.0450	2778.8436	0.7741	1317.0408
N	1.1851	3151.4376	0.9910	1686.0467
Nw	1.1854	3152.1949	0.9912	1686.3547

Table 3. Performance comparison of statistical models. Notes: A - ARIMA model; E - Exponential Smoothing; N - naive model; AE - ARIMA + Exponential smoothing (average); w - rolling window.

time series	the total number of alerts		port 445/TCP	
best NN/statistical model	e1d1 MAE	E	GRU MASE	AE
MASE	**0.9166**	1.0319	**0.6210**	0.7661
MAE	**2437.3081**	2744.0770	**1056.5102**	1303.3641

Table 4. Comparison of best models based on neural networks and statistical models on both time series. Notes: NN - neural network; E - Exponential Smoothing; AE - ARIMA + Exponential smoothing (average).

Fig. 1. Graphical comparison of best models based on neural networks (e1d1 with MAE loss) and statistical models (Exponential smoothing) for the total number of alerts.

Finally, we compared the best statistical method (Exponential Smoothing) and the best neural network (e1d1 MAE, respectively GRU MASE). As may be

seen from Tab. 4, in both cases, neural networks have better MASE and MAE metrics.

Fig. 1 shows one step predictions with the best statistical approach and neural network approach for the total number of alerts that are similar to the naive forecasting with drift. In the same way, in Fig. 2 are predictions for port 445/TCP that are way more accurate in the case of the neural network approach.

Fig. 2. Graphical comparison of best models based on neural networks (Gru with MASE loss) and statistical models (combination of ARIMA and Exponential smoothing) for port 445/TCP.

In addition to the above, we also compared the accurate predictions of the best methods from statistical and neural network approaches. For this purpose, we used the Diebold-Marian test [41] and its implementation in the R package *multDM* [42]. If two forecasts have the same accuracy, it represents the null hypothesis (H0). On the other hand, the alternative hypothesis (H1) had the

setting w = "less" (the first forecast is less accurate than the second forecast). Since we leave a 5% uncertainty rate, the p-value to confirm the null hypothesis (H0) should be higher than 0.05.

In our evaluation, we compared all combinations of the statistical and neural methods used in this paper for two cases - "count of all alerts" and "port 445/TCP".We tested the situation that forecasts have the same accuracy (null hypothesis) against the situation that a forecast based on the statistical method is less accurate than the forecast based on the neural network method (alternative hypothesis). For example, in the Diebold-Mariano test, the p-value of comparison of Arima and dense with MAE loss is 0.003633, which is less than 0.05. In this case, an alternative hypothesis was accepted (the null hypothesis was rejected). It means that for the time series marked "Count of all alerts", the forecast based on the statistical method (Arima) is less accurate than the forecast based on the neural method (dense with MAE loss).

	A	Aw	AE	AEw
cnn MAE	3.331×10^{-07}	1.004×10^{-07}	2.355×10^{-06}	8.033×10^{-07}
cnn MASE	2.87×10^{-05}	1.379×10^{-05}	0.0001242	6.342×10^{-05}
cnn MSE	3.794×10^{-07}	1.338×10^{-07}	2.427×10^{-06}	9.099×10^{-07}
dense MAE	0.003633	0.002393	0.01623	0.009665
dense MASE	0.009035	0.005458	0.04873	0.02757
dense MSE	0.0055	0.003181	0.02018	0.0121
e1d1 MAE	1.509×10^{-07}	6.281×10^{-08}	9.142×10^{-07}	3.617×10^{-07}
e1d1 MASE	4.465×10^{-08}	1.45×10^{-08}	3.009×10^{-07}	1.011×10^{-07}
e1d1 MSE	2.301×10^{-08}	7.747×10^{-09}	1.509×10^{-07}	5.148×10^{-08}
gru MAE	2.303×10^{-07}	9.031×10^{-08}	1.721×10^{-06}	6.213×10^{-07}
gru MASE	6.323×10^{-08}	2.475×10^{-08}	5.847×10^{-07}	1.976×10^{-07}
gru MSE	7.88×10^{-08}	3.027×10^{-08}	5.738×10^{-07}	2.036×10^{-07}
lstm MAE	7.155×10^{-08}	2.748×10^{-08}	4.925×10^{-07}	1.827×10^{-07}
lstm MASE	2.239×10^{-08}	8.27×10^{-09}	1.679×10^{-07}	5.845×10^{-08}
lstm MSE	2.393×10^{-08}	1.022×10^{-08}	1.887×10^{-07}	6.967×10^{-08}

Table 5. Results of Diebold-Marino test for 1-step forecasting of the count of all alerts (Part I). Notes: A - Arima; AE - ARIMA + Exponential smoothing (average); w - rolling window.

Results of Diebold-Marino test of statistical methods and neural network methods for 1-step forecasting of the "Count of all alerts" time series are shown in Tab. 5 and Tab. 6. In these tables, in the first column, there are neural networks methods and the other columns contain p-value for the Diebold-Marino test for a couple of the statistic and neural methods. The forecasts based on statistical methods are less accurate than a forecast based on neural network methods in almost all cases. In two cases the combination of the forecasting methods has the same accuracy. These cases are highlighted (bold font) in Tab. 6.

Results of Diebold-Marino test of statistical methods and neural network methods for 1-step forecasting of the port 445/TCP are shown in Tab. 7 and

	E	Ew	N	Nw
cnn MAE	1.124×10^{-05}	4.263×10^{-06}	1.644×10^{-06}	1.629×10^{-06}
cnn MASE	0.0003741	0.0001893	5.355×10^{-06}	5.323×10^{-06}
cnn MSE	1.059×10^{-05}	4.168×10^{-06}	1.061×10^{-06}	1.048×10^{-06}
dense MAE	0.04673	0.02514	5.046×10^{-05}	4.973×10^{-05}
dense MASE	**0.1431**	**0.07876**	0.0003169	0.0003122
dense MSE	0.04932	0.02911	0.0001261	0.0001246
e1d1 MAE	3.876×10^{-06}	1.491×10^{-06}	6.964×10^{-07}	6.9×10^{-07}
e1d1 MASE	1.435×10^{-06}	5.127×10^{-07}	5.35×10^{-07}	5.297×10^{-07}
e1d1 MSE	7.057×10^{-07}	2.531×10^{-07}	3.218×10^{-07}	3.183×10^{-07}
gru MAE	8.847×10^{-06}	3.151×10^{-06}	2.218×10^{-06}	2.19×10^{-06}
gru MASE	3.693×10^{-06}	1.202×10^{-06}	2.027×10^{-06}	1.993×10^{-06}
gru MSE	3.028×10^{-06}	1.051×10^{-06}	1.006×10^{-06}	9.934×10^{-07}
lstm MAE	2.402×10^{-06}	8.777×10^{-07}	7.2×10^{-07}	7.123×10^{-07}
lstm MASE	8.959×10^{-07}	3.085×10^{-07}	4.627×10^{-07}	4.575×10^{-07}
lstm MSE	1.059×10^{-06}	3.633×10^{-07}	6.262×10^{-07}	6.194×10^{-07}

Table 6. Results of Diebold-Marino test for 1-step forecasting of the count of all alerts (Part II). The cases where the p-value is greater than 0.05 are highlighted (bold font). Notes: E - Exponential Smoothing; N - naive method; w - rolling window.

Tab. 8. In these tables, in the first column, there are neural networks methods and the other columns contain p-value for the Diebold-Marino test for a couple of the statistic and neural methods. The forecasts based on statistical methods are less accurate than a forecast based on neural network methods in all cases.

	A	Aw	AE	AEw
cnn MAE	6.793×10^{-07}	7.495×10^{-08}	0.0004185	8.341×10^{-05}
cnn MASE	6.963×10^{-06}	5.428×10^{-07}	0.00301	0.0005788
cnn MSE	4.434×10^{-06}	7.059×10^{-07}	0.003503	0.0008352
dense MAE	2.788×10^{-13}	3.497×10^{-14}	1.722×10^{-08}	9.922×10^{-10}
dense MASE	6.11×10^{-12}	3.202×10^{-13}	1.73×10^{-07}	7.755×10^{-09}
dense MSE	5.697×10^{-14}	1.508×10^{-15}	3.285×10^{-09}	5.847×10^{-11}
e1d1 MAE	2.055×10^{-13}	6.905×10^{-15}	8.538×10^{-09}	7.248×10^{-10}
e1d1 MASE	1.624×10^{-14}	2.595×10^{-16}	8.628×10^{-10}	4.84×10^{-11}
e1d1 MSE	8.302×10^{-15}	1.028×10^{-15}	2.572×10^{-10}	3.39×10^{-11}
gru MAE	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	6.626×10^{-13}	2.885×10^{-14}
gru MASE	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
gru MSE	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	1.712×10^{-12}	1.165×10^{-13}
lstm MAE	4.18×10^{-11}	1.474×10^{-12}	2.657×10^{-07}	2.998×10^{-08}
lstm MASE	1.245×10^{-10}	2.13×10^{-11}	6.396×10^{-07}	1.287×10^{-07}
lstm MSE	8.967×10^{-11}	1.966×10^{-11}	6.882×10^{-07}	1.398×10^{-07}

Table 7. Results of Diebold-Marino test for 1-step forecasting of the port 445/TCP (Part I). Notes: A - Arima; AE - ARIMA + Exponential smoothing (average); w - rolling window.

	E	Ew	N	Nw
cnn MAE	0.0004185	8.341×10^{-05}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
cnn MASE	0.00301	0.0005788	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
dense MAE	1.722×10^{-08}	9.922×10^{-10}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
dense MASE	1.73×10^{-07}	7.755×10^{-09}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
dense MSE	3.285×10^{-09}	5.847×10^{-11}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
e1d1 MAE	8.538×10^{-09}	7.248×10^{-10}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
e1d1 MASE	8.628×10^{-10}	4.84×10^{-11}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
e1d1 MSE	2.572×10^{-10}	3.39×10^{-11}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
gru MAE	6.626×10^{-13}	2.885×10^{-14}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
gru MASE	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
gru MSE	1.712×10^{-12}	1.165×10^{-13}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
lstm MAE	2.657×10^{-07}	2.998×10^{-08}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
lstm MASE	6.396×10^{-07}	1.287×10^{-07}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$
lstm MSE	6.882×10^{-07}	1.398×10^{-07}	$< 2.2 \times 10^{-16}$	$< 2.2 \times 10^{-16}$

Table 8. Results of Diebold-Marino test for 1-step forecasting of the port 445/TCP (Part II). Notes: Notes: E - Exponential Smoothing; N - naive method; w - rolling window.

These calculations confirm our results expressed by MAE and MASE measures described above.

6 Conclusion and future works

Within the paper, we focused on NSSA forecasting. For this purpose, we used two-time series (the total number of alerts and alerts related to the services running on port 445/TCP). These time series represent two categories of time series for the area of NSSA forecasting (well predictable time series, and unpredictable time series) [8]. This paper aimed to analyse the impact of loss function on the accuracy of NSSA forecasting based on neural networks. According to the obtained results, we found that the loss function has an effect and the MAE and MASE loss function give comparable results. At the same time, we compared the best neural networks and the best statistical methods. According to the MASE and MAE metrics, we can state that neural networks are more accurate for NSSA forecasting. As part of future works, we would like to focus on NSSA forecasting on time series created from other security alerts (obtained by a platform other than the Warden system).

Acknowledgment. This research is funded by the VVGS projects under contract No. VVGS-PF-2020-1423, VVGS-PF-2020-1427, and VVGS-PF-2021-1792 and Slovak Research and development agency project under contract No. APVV-17-0561.

References

1. Cho, J.H., Sharma, D.P., Alavizadeh, H., Yoon, S., Ben-Asher, N., Moore, T.J., Kim, D.S., Lim, H., Nelson, F.F.: Toward proactive, adaptive defense: A survey on moving target defense. IEEE Commun. Surv. Tutor **22**(1) (2020) 709–745
2. Carle, G., Dressler, F., Kemmerer, R.A., Koenig, H., Kruege, C., Laskov, P.: Network attack detection and defense. In: Manifesto of the Dagstuhl Perspectives Workshop, March. (2008) 2–6
3. Bass, T., et al.: Multisensor data fusion for next generation distributed intrusion detection systems. In: Proceedings of the IRIS National Symposium on Sensor and Data Fusion. Volume 24., Citeseer (1999) 24–27
4. Jiang, Y., Li, C.h., Yu, L.s., Bao, B.: On network security situation prediction based on rbf neural network. In: 2017 36th Chinese Control Conference (CCC), IEEE (2017) 4060–4063
5. Shang, L., Zhao, W., Zhang, J., Fu, Q., Zhao, Q., Yang, Y.: Network security situation prediction based on long short-term memory network. In: 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE (2019) 1–4
6. Husák, M., Bartoš, V., Sokol, P., Gajdoš, A.: Predictive methods in cyber defense: Current experience and research challenges. Future Generation Computer Systems **115** (2021) 517–530
7. Sokol, P., Gajdoš, A.: Prediction of attacks against honeynet based on time series modeling. In: Proceedings of the Computational Methods in Systems and Software, Springer (2017) 360–371
8. Sokol, P., Staňa, R., Gajdoš, A., Pekarčík, P.: Network security situation awareness forecasting based on statistical approach and neural networks. Logic Journal of the IGPL (2022)
9. Liu, Y., Sarabi, A., Zhang, J., Naghizadeh, P., Karir, M., Bailey, M., Liu, M.: Cloudy with a chance of breach: Forecasting cyber security incidents. In: 24th USENIX Security Symposium 15. (2015) 1009–1024
10. Okutan, A., Werner, G., McConky, K., Yang, S.J.: Poster: Cyber attack prediction of threats from unconventional resources (capture). In: 24th ACM Conference on Computer and Communications Security. (2017) 2563–2565
11. Werner, G., Yang, S., McConky, K.: Time series forecasting of cyber attack intensity. In: Proceedings of the 12th Annual Conference on Cyber and Information Security Research, ACM (2017) 1–3
12. Werner, G., Yang, S., McConky, K.: Leveraging intra-day temporal variations to predict daily cyberattack activity. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE (2018) 58–63
13. Werner, G., Okutan, A., Yang, S., McConky, K.: Forecasting cyberattacks as time series with different aggregation granularity. In: 2018 IEEE International Symposium on Technologies for Homeland Security (HST), IEEE (2018) 1–7
14. Qi, Y., Shang, W., He, X.: A combined prediction method of industrial internet security situation based on time series. In: Proceedings of the 2019 the 9th International Conference on Communication and Network Security. (2019) 84–91
15. Zhan, Z., Xu, M., Xu, S.: Predicting cyber attack rates with extreme values. IEEE Transactions on Information Forensics and Security **10**(8) (2015) 1666–1677
16. Pillai, T.R., Palaniappan, S., Abdullah, A., Imran, H.M.: Predictive modeling for intrusions in communication systems using garma and arma models. In: 2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW), IEEE (2015) 1–6

17. Zhang, H., Huang, Q., Li, F., Zhu, J.: A network security situation prediction model based on wavelet neural network with optimized parameters. Digital Communications and Networks **2**(3) (2016) 139–144
18. He, F., Zhang, Y., Liu, D., Dong, Y., Liu, C., Wu, C.: Mixed wavelet-based neural network model for cyber security situation prediction using modwt and hurst exponent analysis. In: International Conference on Network and System Security, Springer (2017) 99–111
19. Feng, W., Wu, Y., Fan, Y.: A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit. International Journal of Intelligent Computing and Cybernetics (2018)
20. Goyal, P., Hossain, K., et al.: Discovering signals from web sources to predict cyber attacks. arXiv preprint arXiv:1806.03342 (2018)
21. Fang, X., Xu, M., Xu, S., Zhao, P.: A deep learning framework for predicting cyber attacks rates. EURASIP Journal on Information Security **2019**(1) (2019) 1–11
22. Lavrova, D., Zegzhda, D., Yarmak, A.: Using gru neural network for cyber-attack detection in automated process control systems. In: 2019 IEEE International Black Sea Conference on Communications and Networking, IEEE (2019) 1–3
23. Filonov, P., Kitashov, F., Lavrentyev, A.: Rnn-based early cyber-attack detection for the tennessee eastman process. arXiv preprint arXiv:1709.02232 (2017)
24. Filonov, P., Lavrentyev, A., Vorontsov, A.: Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. arXiv preprint arXiv:1612.06676 (2016)
25. Kacha, P., Kostenec, M., Kropacova, A.: Warden 3: Security event exchange redesign. In: 19th International Conference on Computers: Recent Advances in Computer Science. (2015)
26. Kacha, P.: Idea: security event taxonomy mapping. In: 18th International Conference on Circuits, Systems, Communications and Computers. (2014)
27. Pekarčík, P., Gajdoš, A., Sokol, P.: Forecasting security alerts based on time series. In: International Conference on Hybrid Artificial Intelligence Systems, Springer (2020) 546–557
28. Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V.: An innovative neural network approach for stock market prediction. The Journal of Supercomputing **76**(3) (2020) 2098–2118
29. Chen, K., Zhou, Y., Dai, F.: A lstm-based method for stock returns prediction: A case study of china stock market. In: 2015 IEEE international conference on big data (big data), IEEE (2015) 2823–2824
30. Kim, T., Kim, H.Y.: Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. PloS one **14**(2) (2019) 1–23
31. Fu, R., Zhang, Z., Li, L.: Using lstm and gru neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE (2016) 324–328
32. Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J.: Lstm network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems **11**(2) (2017) 68–75
33. Brownlee, J.: Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery (2018)
34. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
35. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)

36. Brockwell, P.J., Davis, R.A.: Introduction to time series and forecasting. Springer (2016)
37. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International journal of forecasting **22**(4) (2006) 679–688
38. Abadi, M., Agarwal, A., et al.: Tensorflow: Large-scale machine learning on heterogeneous systems, software available from tensorflow. org (2015). URL https://www. tensorflow. org (2015)
39. Biewald, L.: Experiment tracking with weights and biases (2020) Software available from wandb.com.
40. Hyndman, R.J., Khandakar, Y., et al.: Automatic time series for forecasting: the forecast package for R. Number 6. Monash University, Department of Econometrics and Business Statistics (2007)
41. Diebold, F.X., Mariano, R.S.: Comparing predictive accuracy. Journal of Business & economic statistics **20**(1) (2002) 134–144
42. Drachal, K.: multmdm: Multivariate version of the diebold-mariano test. (2018)