UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

PREDIKCIA ČASOVÝCH RADOV V OBLASTI POČÍTAČOVEJ BEZPEČNOSTI

2023 **Barbora Feďová**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

PREDIKCIA ČASOVÝCH RADOV V OBLASTI POČÍTAČOVEJ BEZPEČNOSTI

BAKALÁRSKA PRÁCA

Študijný program:	Aplikovaná informatika
Pracovisko (katedra/ústav):	Ústav informatiky
Vedúci bakalárskej práce:	RNDr. Richard Staňa
Konzultant bakalárskej práce:	doc. RNDr. JUDr. Pavol Sokol, PhD .

Košice 2023

Barbora Feďová

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Barbora Feďová

Študijný program: aplikovaná informatika (jednoodborové štúdium, bakalársky I. st., denná forma)

Študijný odbor: Informatika

Typ záverečnej práce: Bakalárska práca

Jazyk záverečnej práce: slovenský

Sekundárny jazyk: anglický

Názov: Predikcia časových radov v oblasti počítačovej bezpečnosti

Názov EN: Time series forecasting in field of computer security

Cieľ: 1. Analyzovať existujúce prístupy predikcie časových radov v oblasti počítačovej bezpečnosti.
2. Implementovať model neurónovej siete na predikciu časových radov založený na modeli Transformer.
3. Porovnať dosiahnuté výsledky s existujúcimi výsledkami.

Literatúra:
1) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez,A. N., ... & Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.
2) Pavol, S., Stana, R., Gajdos, A., Patrik, P.: Network security awarenessforecasting based on statistical approach and neural networks. Logic Journal of IGPL (Submitted)

Vedúci: RNDr. Richard Staňa

Konzultant: doc. RNDr. JUDr. Pavol Sokol, PhD.

Oponent: doc. RNDr. Csaba Török, CSc.

Ústav : ÚINF – Ústav informatiky

Riaditeľ ústavu: doc. RNDr. Ondrej Krídlo, PhD.

Dátum schválenia: 15.05.2023

Poďakovanie

Chcela by som týmto úprimne poďakovať školiteľovi mojej záverečnej bakalárskej práce RNDr. Richardovi Staňovi a konzultantovi práce doc. RNDr. JUDr. Pavlovi Sokolovi, PhD., za pomoc pri spracovávaní práce, ich odborné rady, trpezlivosť a rýchlu komunikáciu. Moja vďaka patrí aj môjmu manželovi, ktorý venoval dlhé hodiny prechádzkam s našou dcérou Elizabetou. Len vďaka tomu som mala čas venovať sa tejto práci

Abstrakt

Existujúcich prístupov na predikciu časových radov dnes poznáme niekoľko. Hneď na úvod v práci definujeme základné pojmy predikcie a jej známych prístupov. Zameriavame sa na predikciu pomocou neurónových sietí v oblasti informačnej bezpečnosti. Vzhľadom na stále pribúdajúce nové technológie je potrebné skúmať situáciu v sieti a navrhovať nové modely, ktoré sa svojím rozšírením vedia vyrovnať s novými okolnosťami v sieti, resp. dosahujú lepšie výsledky v predikcii v porovnaní s inými modelmi. Na základe uvedeného navrhneme nový model založený na sieti typu Transformer určený na predikciu časových radov v oblasti bezpečnosti na internete a porovnáme jeho výsledky s inými modelmi na rovnakých dátach.

Kľúčové slová: *predikcia časových radov, neurónová sieť, Transformer, počítačová bezpečnosť*

Abstract

Today is known a lots of approaches for time series forecasting. In the introduction we define basic terms of forecasting and known approaches. We mainly focus on forecasting information security with neural network. Due to increasing new technologies, it is necessary to examine the situation in the network and design new models that can deal with new circumstances in the network or they have better forcasting results compared to other models. This is the next goal of this bachelor thesis. Implement a new model based on model Transformer intended for time series forecasting in the field of network security and compare results with another one on the same data.

Key words: *time series forecasting, neural network, Transformer, computer security*

Úvod

Strojové učenie dnes predstavuje rozsiahlu tému vďaka jeho širokému využitiu. Je mu vlastná aj schopnosť predpovedania budúcich javov, resp. budúceho správania sa na základe súboru metód a samoučiacich sa algoritmov. Výnimočnosť daných algoritmov spočíva v tom, že sa vedia modifikovať samy, bez zásahu človeka. Inak povedané, umožňujú počítačovým systémom automaticky sa učiť bez toho, aby museli byť zakaždým programované. Dnes má strojové učenie spolu s umelou inteligenciou nenahraditeľné miesto v oblasti informatiky, čo v konečnom dôsledku zefektívňuje a mení rôzne ďalšie oblasti.

V práci sa budeme venovať predikcii časových radov, ktoré majú rozsiahlu využiteľnosť v oblasti kybernetickej bezpečnosti. Vo všeobecnosti táto predikcia vôbec nie je jednoduchá. Predikcia časových radov je zložitejšia o to viac tým, že sa kladie dôraz na poradie a časovú závislosť medzi údajmi pri ich spracovávaní. V súčasnosti existuje mnoho modelov na predikciu bezpečnostných udalostí, no s pribúdajúcimi novými technológiami je potrebné zlepšovať aj modely na predikciu.

Dnes už mnohé oblasti využívajú spracovávanie konkrétnych údajov do časových radov a ich následnú predikciu. To im prináša vyvarovanie sa prípadným stratám alebo hrozbám. To isté platí aj pri bezpečnosti na internete. Spracovanie konkrétnych údajov a ich následná predikcia môže pomôcť zlepšiť bezpečnosť na internete tým, že umožňuje identifikovať a predvídať nebezpečné situácie a správať sa tak preventívne. Tieto predikcie by mohli identifikovať aj také hrozby, ktoré by sme inak prehliadli. Umožňuje teda identifikáciu anomálií, ktoré môžu byť indikátormi útokov. Napríklad keď sa určité udalosti vyskytnú v neočakávanom poradí, môže to znamenať, že niekto sa snaží napadnúť zariadenie alebo sieť. Vďaka tomu je možná rýchlejšia a účinnejšia príprava a následné reagovanie na dané hrozby. Analýza časových radov nám poskytuje aj predstavu o tom, aké sú pravdepodobné budúce udalosti. Napríklad ak sú známe isté špecifické typy útokov, môžu sa použiť časové rady na predpovedanie týchto útokov. Znovu to pomáha pri príprave na zabezpečenie systémov a sietí. Ak sa isté útoky vyskytujú pravidelne, môže sa vyvinúť stratégia, ako im predchádzať a minimalizovať tak ich dôsledky. Predikcia tiež napomáha analyzovať správanie sa používateľov na internete a nájsť tak ich potencionálne nebezpečné aktivity. Rovnako môže byť užitočná pri testovaní účinnosti bezpečnostných opatrení. Keď sa implementuje nový

bezpečnostný nástroj, môže sa použiť analýza časových radov na monitorovanie jeho účinnosti a zisťovanie, či sa podarilo zmierniť riziká bezpečnostných hrozieb.

Jedným z cieľov tejto práce je analyzovanie existujúcich prístupov predikcie časových radov v oblasti počítačovej bezpečnosti či popis modelov a porovnanie výhod a nevýhod každého prístupu. Väčšinou sa používajú rôzne kombinácie, čo je tiež zahrnuté v prvej časti práce.

Ďalším z cieľov je implementácia vlastného modelu neurónovej siete na predikciu časových radov. Predikcia založená na modeli Transformer v tejto oblasti zatiaľ nie je veľmi známa a prináša nový pohľad na samotnú predikciu.

Výsledky z tejto predikcie je dôležité dôkladne vyhodnotiť, vďaka čomu je možné ich následne porovnať s výsledkami z iných predikcií. Posledným hlavným cieľom práce bude práve uvedené porovnanie dosiahnutých výsledkov s existujúcimi výsledkami a vyhodnotenie, či náš nový model priniesol nejaké zlepšenie pri predikcii časových radov, resp. ktoré modely sú v akých prípadoch lepším variantom.

1 Úvod do predikcie

Predikcia je predpovedaná hodnota alebo očakávaný výsledok, ktorý je odhadnutý na základe dostupných dát, modelov alebo algoritmov. To znamená, že vieme určiť, ako sa budú vyvíjať hodnoty vo vybranom prostredí, a pomocou ďalších metrík aj kvalitu predikcie. Predikovať môžeme napríklad v textovom súbore predikciu ďalšieho slova, obrazová predikcia alebo údajová predikcia s dôležitými hodnotami predstavujúcimi napríklad bezpečnostné udalosti, zisky spoločnosti a iné. Tie sa však pre lepšiu schopnosť predikovania spracovávajú a ukladajú v tzv. časových radoch.

„**Časový rad** je súbor pozorovaní x_t, pričom každý bol zaznamenaný v špecifickom čase t“ [58]. Pre predikciu časových radov je potrebné veľké množstvo hodnôt. Výhodou časových radov je, že si ich vieme transformovať na graf, pričom ukazovateľom hodnôt v grafe je počet konkrétnej udalosti a čas. Takéto dáta sú jednoduchšie spracovateľné aj na konečnú vizualizáciu samotnej predikcie, čím si vieme jednoducho odsledovať presnosť a kvalitu predikcie [1].

Mohlo by sa zdať, že predikcia je len na predvídanie konštantného prostredia a nie je možné predvídať meniace sa prostredie. No v skutočnosti je každé prostredie premenlivé a dobrý predpovedný model zachytáva spôsob, akým sa veci menia. Prognózy len zriedka predpokladajú, že prostredie je nemenné [1]. „Modely určené na predikciu poskytujú spoľahlivé odpovede na zložitejšie problémy, skúmajú nové druhy problémov a prinášajú odpovede v reálnom čase na problémy v prostredí s meniacimi sa informáciami“ [40].

Predvídateľnosť udalosti alebo množstva závisí od [1]:

1. schopnosti porozumenia faktorom, ktoré k tomu prispievajú;
2. počtu údajov, ktoré máme k dispozícii;
3. miery podobnosti budúcnosti s minulosťou;
4. účinku predpovedí (resp. či predpovede môžu ovplyvniť vec, ktorú sa snažíme predpovedať).

„Základom správnej predikcie pri extrapolácii trendovej funkcie je vhodná voľba analytickej funkcie charakterizujúcej doterajší vývoj sledovaného ukazovateľa“ [2]. To znamená, že pri vytváraní predikcie je veľmi dôležité vykonať analýzu doterajších hodnôt. Treba vedieť, či zozbierané údaje z minulosti sa diali na základe určitého vzorca alebo boli celkom náhodné. Náhodné udalosti obsiahnuté v údajoch je potrebné rozlíšiť

a ignorovať. Naopak skutočný vzorec vychádzajúci zo zozbieraných údajov by sme mali brať do úvahy pri vytváraní modelu na predikciu. Zahrnutie náhodných udalostí bude viesť k nesprávnemu natrénovaniu modelu na predikcie, čo v konečnom dôsledku vedie k nepresným a chaotickým predikciám.

1.1 Delenie predikcií z pohľadu časového rozpätia

Predikcia je známa a obľúbená v rôznych oblastiach, napríklad pri podnikaní sa využíva pri plánovaní výroby, dopravy a personálu. Nie vždy sú modely na predikciu nastavené správne. Vo všeobecnosti si organizácie rozlišujú a vyžadujú tri typy predikcií v závislosti od konkrétnej aplikácie, a to [1]:

- krátkodobé predikcie,
- strednodobé predikcie a
- dlhodobé predikcie.

Krátkodobé predikcie sú efektívne a využívané hlavne na plánovanie personálu, výroby, dopravy alebo aj dopytu. Vytvárame predikcie pre krátky časový interval, čo je využiteľné pre odvetvia a prípady, kde sa údaje môžu rýchlo zmeniť, a preto chceme brať do úvahy a predikovať krátky časový interval. Môžu to byť hodiny, dni alebo týždne dopredu. Majú vyššiu dôveryhodnosť ako dlhodobé predikcie, pretože sa zameriavajú na blízku budúcnosť a obvykle majú k dispozícii viac dostupných údajov a majú tendenciu byť presnejšie.

Strednodobé predikcie sú potrebné skôr na budúce požiadavky na zdroje, najatie personálu, nákup surovín, strojov a zariadení. Tieto udalosti ovplyvňujú ďalšie udalosti z dlhodobejšieho hľadiska ako boli krátkodobé predikcie [1]. Zameriava sa na taktické rozhodovanie v priebehu mesiacov, niekedy aj pár rokov. V porovnaní s krátkodobou predikciou sú náchylnejšie na vplyvy nepredvídateľných udalostí, ako sú ekonomické krízy alebo politické zmeny.

Dlhodobá predikcia sa stretáva s námietkou, že je prakticky nemožné predpovedať s presnosťou, čo sa stane o niekoľko rokov v budúcnosti. Čím dlhšia je predikovaná doba, tým viac sa zvyšuje neistota v predikované hodnoty. Avšak prinajmenšom dlhodobá predikcia a miera jej presnosti môže poukazovať na riziká pri presadzovaní zvolenej stratégie a vďaka tomu pomáha lepšie sa rozhodnúť pre jednu

z dostupných stratégií [38]. Využívajú sa hlavne na strategické plánovanie, kde sa musia brať do úvahy aj trhové príležitosti, environmentálne faktory a vnútorné zdroje [1].

Je dôležité zvážiť časové rozpätie, teda na aký dlhý čas chceme predikovať nejakú udalosť. Podľa toho, či chceme predikovať na mesiac dopredu, rok či desať rokov, budeme si vyberať aj typ modelu na predikciu. Je to prvotná a základná otázka. Každý typ predikcie má svoje vlastné využitie aj obmedzenia.

V súčasnosti sa čoraz viac stretávame s možnosťou automatických predikcií. Uľahčuje to spôsob fungovania napríklad prevádzky v prípade, že chceme vytvárať predikcie v krátkych časových intervaloch. Teda vzniká ďalšia dôležitá otázka, ktorá by mala byť zodpovedaná pred výberom modelu. Ak chceme predikcie v krátkych časových intervaloch, tak je dobré zvážiť automatický systém, ktorý nie je potrebné často manuálne nastavovať. V dlhších časových intervaloch je vhodnejšia metóda, ktorá si vyžaduje manuálne nastavovanie pre presnejšie výsledky. Je to najmä pre množstvo údajov a rizika, že v priebehu času sa môže prostredie, údaje alebo potreby predikcie zmeniť. V automatickom systéme by sme v takomto prípade mohli prehliadnuť zmeny potrieb pre robustnosť predikcie, resp. bolo by časovo nákladnejšie meniť automatický systém podľa nových požiadaviek a potrieb.

Dôležitou informáciou je aj časový horizont, ktorý označuje, za aký čas boli zhromažďované konkrétne údaje. Môžu byť v rôznych intervaloch, napríklad v päťminútových, denných, týždenných, ročných intervaloch.

Konkrétnym príkladom pre vyhovujúce údaje pre predikciu sú ročné zisky spoločnosti, štvrťročné výsledky predaja, hodinová spotreba elektriny, týždenný maloobchodný predaj, mesačné zrážky atď.. Na základe takýchto údajov teda vieme predikovať, aké hodnoty budeme dosahovať v budúcnosti.

Základné, najjednoduchšie metódy, nie sú dostatočne robustné na to, aby vedeli brať do úvahy aj faktory, ktoré môžu ovplyvniť hodnoty atribútov v budúcnosti. Ide napríklad o aktivitu konkurencie, zmenu ekonomických podmienok a podobne [1].

Primárne otázky dôležité pri výbere modelu možno zhrnúť do piatich základných krokov, a to [1]:

- definícia problému,
- zhromaždenie informácií,
- predbežná analýza,
- výber modelu a jeho tréning,

- použitie a vyhodnotenie modelu na predikciu.

Prvým krokom je **definícia problému**. Dôkladné definovanie problému si vyžaduje pochopenie spôsobu, akým sa budú prognózy používať. Teda potrebné je aj venovať čas ľuďom, ktorí sa budú podieľať na zhromažďovaní údajov a údržbe databáz. Druhým krokom je **zhromaždenie informácií**. Často sa vyskytuje problém nedostatku historických údajov, keď nie je možné natrénovať navrhnutý model. Pri **predbežnej analýze** je potrebné začať pohľadom na dáta. Vzniká viacero otázok, ktoré treba zodpovedať. Je sezónnosť dôležitá? Sú v údajoch hodnoty, ktoré je potrebné vysvetliť odborníkmi? Zachádzame teda do väčších detailov a snažíme sa analyzovať všetko, čo sa týka predikovania. Nasledujúcim krokom je **výber a tréning modelu**. Štandardne sa vyberá viac možných modelov a z nich sa na základe ďalších kritérií zvolí ten najvhodnejší. Selektovaný je na základe dostupnosti údajov, sily vzťahov a spôsobu, akým sa majú prognózy používať. Posledným krokom je **použitie a vyhodnotenie modelu na predikciu**. Konečný model použijeme na tvorbu prognóz. Bolo vyvinutých aj niekoľko metód, ktoré pomáhajú pri hodnotení presnosti predpovedí. V praxi sa napriek dôkladnej príprave nachádza mnoho praktických problémov, napríklad ako zvládnuť chýbajúce hodnoty alebo ako sa vyrovnať s krátkymi časovými radmi [1].

1.2 Základné pojmy z oblasti predikcie časových radov.

Je potrebné definovať aj základné pojmy, ktoré sú úzko späté s časovými radmi. Zlé vysvetlenie alebo nepochopenie týchto pojmov by nás mohlo doviesť do omylu a výberu nesprávnej metódy na predikciu. Sú to pojmy **trend, sezónnosť** a **cyklickosť**.

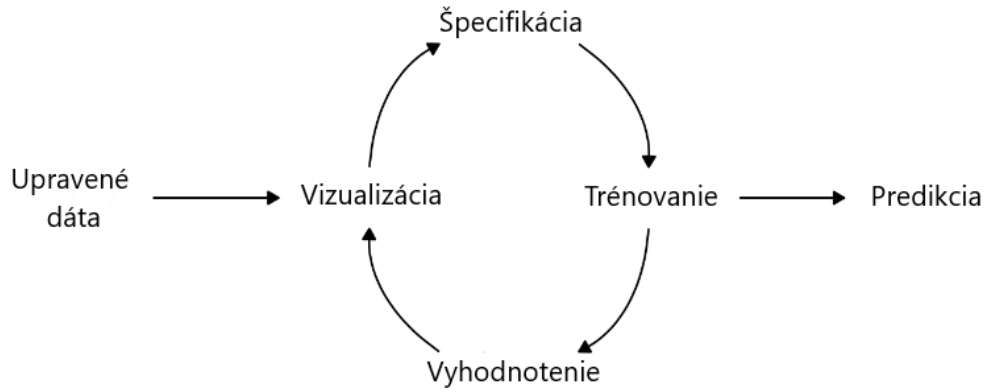
Trendom sa označuje dlhodobý nárast alebo pokles údajov. Trend môže byť rastúci, klesajúci alebo stagnujúci [2]. Môže dôjsť aj k zmene smeru, keď z rastúceho trendu prejdeme na klesajúci trend [1]. Trend je jednoduché identifikovať pomocou počítačovej analýzy údajov, kde sa používajú štatistické metódy ako je prispôsobenie trendovej čiary alebo použitie autokorelačnej funkcie [39].

Sezónnosť nastáva, keď je časový rad ovplyvnený sezónnymi faktormi, ako je napríklad čas v roku alebo deň v týždni. Hlavným znakom je, že má vždy pevné a známe obdobie [1]. Môžu to byť aj štátne sviatky, prázdniny. Je dôležité predpovedať zmeny v časových radoch, ktoré môžeme pripísať sezónnosti. Tieto informácie neposkytuje tradičná regresná analýza a môže to byť dôležitým bodom pri vyberaní

najvhodnejšieho prognostického modelu. Sezónnosť v údajoch môže taktiež zakrývať prítomnosť trendu alebo cyklické vzorce [39].

Cyklickosť nastáva, keď údaje vykazujú vzostupy a poklesy, ktoré nemajú pevnú frekvenciu. Tieto výkyvy sú zvyčajne spôsobené ekonomickými podmienkami [1]. Na rozdiel od sezónnosti cyklickosť pretrváva dlhšie a nemusia byť jednoznačne predvídateľné.

Taktiež je vhodná úprava historických údajov, ktorá často vedie k jednoduchšiemu časovému radu. Poznáme štyri druhy úprav: kalendárne úpravy, populačné úpravy, inflácie a matematické transformácie [1]. Účelom týchto úprav je vytvorenie konzistentnejšieho vzoru v rámci celého súboru údajov. V konečnom dôsledku sa takto upravené dáta ľahšie modelujú a vedú k presnejším prognózam.

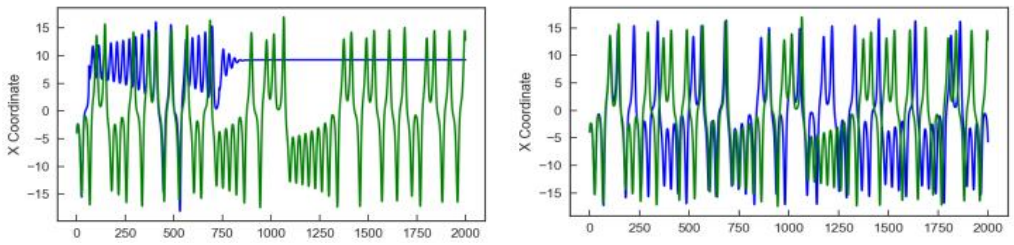


Obr. 1: Proces vytvárania predikcií. Prevzaté z: [1]

Prvým základným krokom pri procese vytvárania predikcií pre údaje časových radov, ako sme vyššie spomínali, je **príprava dát**. Zahŕňa to načítanie dát, zistenie chýbajúcich údajov, filtrovanie časových radov a ďalšie úlohy predbežného spracovania. Každý model má rôzne požiadavky na údaje, aj preto je dôležité hneď na začiatku si pripraviť vhodné dáta a cez ne dôjsť aj k pochopeniu funkcií modelu. **Vizualizácia údajov** je taktiež základným krokom k pochopeniu údajov. Pri zobrazení dát vieme analyzovať bežný vzor a následne určiť vhodný model. Existuje mnoho rôznych modelov na vytvorenie predikcií časových radov. Práve preto je **špecifikácia vhodného modelu** nevyhnutná pre vytvorenie vhodných predpovedí. Časť zozbieraných údajov sa nazývajú tréningové dáta. Na týchto dátach následne prebieha samotný **tréning modelu**. Pomocou tréningu vieme odhadnúť jednu alebo viac špecifikácií modelu [1]. Prostredníctvom

diagnostických nástrojov dokážeme **vyhodnotiť výkon modelu**. To znamená, že zistíme, ako dobre pracoval s našimi údajmi. Takisto sem zahŕňame aj merania presnosti, ktoré nám umožňujú porovnávať jeden model s iným. Konečnou fázou je vytvorenie samotnej predikcie. Teda až so špecifikovaným, odhadnutým a skontrolovaným vhodným modelom vytvoríme prognózu.

Pre lepšiu predstavu, ako by mala, resp. nemala vyzerať predikcia, zobrazujeme predikcie dvoch modelov, ktoré sú výsledkom článku [41]. Na obrázku č. 2 je znázornený príklad nepresnej predikcie, kde si vieme ľahko porovnať výsledky predikcie (modré hodnoty) a očakávané hodnoty (zelené hodnoty), ktoré sú v istých bodoch absolútne odlišné od predikcie. Táto predikcia by nám v skutočnosti veľmi nepomohla odhadnúť budúce hodnoty. Na obrázku č. 3 je zobrazený príklad presnej predikcie, kde pri výbere modelu dbali na dôkladnú prípravu, a teda aj výsledok zodpovedá očakávaniam.



Obr. 2 (vľavo), obr. 3 (vpravo): Príklad nepresnej a presnej predikcie. Prevzaté z: [41]

2 Existujúce prístupy predikcie časových radov

V súčasnosti poznáme viacero možných spôsobov na predikciu časových radov. Každý model určený na predikciu má isté výhody oproti ostatným a zároveň aj svoje negatíva. Podľa okolností, ktoré sme už vyššie spomínali, si vyberáme vhodný model na vybrané dáta. Existuje viacero prístupov, no spomenieme tie najpoužívanejšie.

Takým známejším a nie dávnym návrhom je **metóda Prophet**. Tento model predstavila spoločnosť Facebook [37] pôvodne na predpovedanie denných údajov s týždennou a ročnou sezónnosťou, navyše s efektami sviatkov. Neskôr bola rozšírená natoľko, aby pokrývala viacero typov sezónnych údajov. Využiteľnosť nachádza pri časových radoch, ktoré majú silnú sezónnosť a niekoľko sezón historických údajov [1].

Donedávna bolo zvažované, že jedným z obmedzení modelov je to, že ukladajú jednosmerný vzťah. To znamená, že premenná prognózy je ovplyvnená predikovanými premennými, no nie naopak. Poznáme však prípady, v ktorých by sa mal pripustiť aj opačný postup, teda keď sa všetky premenné navzájom ovplyvňujú. Vtedy vieme využiť **model vektorovej autoregresie** [1].

K najpoužívanejším modelom na predikciu časových radov patrí určite model **ARIMA** (autoregressive integrated moving average). Tento model je označený ako ARIMA(p, d, q), kde „p“ je počet autoregresívnych členov, „d“ predstavuje počet rozdielov a „q“ je počet kĺzavých priemerov. Autoregresívne modely predpokladajú, že predchádzajúce hodnoty tvoria lineárnu funkciu a že každé jedno porovnanie obsahuje náhodnú zložku a lineárnu kombináciu predchádzajúcich pozorovaní [42]. Z tohto modelu vznikli mnohé ďalšie pridaním rôznych vylepšených vlastností. Napríklad model SARIMA, SARIMAX a mnohé iné.

Ďalším možným riešením predikcie by mohlo byť generovanie nových časových radov pomocou **bootstrapu**, ktoré sú podobné našim pozorovaným radom. Dôvodom vzniku tohto prístupu je pokus o zlepšenie presnosti predikcie. Podstatou je, že ak vytvoríme predikcie z každého dodatkového časového radu a spriemerujeme výsledné predpovede, získame lepšie predpovede, ako keby sme priamo predpovedali pôvodný časový rad. Toto sa nazýva „**bagging**“, čo je skratka pre „bootstrap aggregating“ [1].

Známym riešením je aj **exponenciálne vyhladzovanie** (exponential smoothing). Podobne ako model ARIMA je aj tento model lineárny vzhľadom na to, že budúce hodnoty sú obmedzené na lineárne funkcie minulých údajov [42]. Najjednoduchší model sa dá popísať nasledovne:

$$S_0 = x_0;$$

$$S_t = \alpha x_t + (1 - \alpha)S_{t-1},\ t > 0,$$

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

Asi najpoužívanejším spôsobom predikcie časových radov je dnes využitie **neurónových sietí**. Ide o prognostické metódy, ktoré patria do oblasti umelej inteligencie. Sú postavené na základe vlastností biologických nervových systémov. Vytvárajú zložité nelineárne vzťahy medzi výstupnou premennou a skupinou premenných zobrazujúcich údaje z minulosti. Napriek zložitosti pochopenia a zostavenia modelu sú veľmi obľúbené pre široké využitie a presnosť pri predikciách.

2.1 Význam predikcie v oblasti bezpečnosti na internete

Bezpečnosť v počítačovej sieti je dnes rozšírená téma vzhľadom na to, že s pribúdajúcimi technológiami pribúdajú aj nové bezpečnostné hrozby. Nové bezpečnostné hrozby a nárast existujúcich bezpečnostných hrozieb zvyšuje množstvo bezpečnostných útokov a incidentov. Keďže všetky aktivity v rámci operačných systémov sú zaznamenávané, narastá aj množstvo udalostí, ktoré sú ukladané a potencionálne môžu súvisieť s informačnou a kybernetickou bezpečnosťou organizácie. Bezpečnostné udalosti môžeme definovať aj ako "výstrahu, čo je správa o tom, že bola zistená zaujímavá udalosť, ktorá obvykle obsahuje informácie o nezvyčajnej aktivite" [59]. Takéto udalosti môžu zahŕňať pokusy o neoprávnený prístup k dátam, šírenie malware, prístup z neautorizovaných zariadení alebo neobvyklé aktivity v sieti. Zvyčajne sa zaznamenávajú pomocou bezpečnostných nástrojov a technológií, ktoré sledujú a analyzujú sieťovú prevádzku a dáta. Môžu identifikovať podozrivé aktivity a upozorniť správcov siete na potencionálne rizikové udalosti. Správne a včasné riešenie bezpečnostných udalostí je kľúčové pre zabezpečenie bezpečnosti informačných systémov a ochranu dát pred stratou alebo krádežou.

Práve preto je dôležité aj tzv. **sieťové situačné povedomie** (Network Security Situation Awareness). Zaraďujeme tam zbieranie a interpretáciu informácií o bezpečnostných hrozbách, aktuálnych udalostiach v počítačovej sieti, ich následnú analýzu a predikciu a taktiež posúdenie frekvencie útokov či odhad miery ohrozenia siete. Dalo by sa to zhrnúť do troch základných bodov, a to informovanosť o situácii v sieti, hodnotenie hrozieb a posúdenie situácie v sieti [27]. So zlepšením sieťového situačného povedomia sa teda zlepšuje aj schopnosť organizácií alebo jednotlivcov predchádzať,

detegovať a rýchlejšie reagovať na bezpečnostné hrozby v sieti, čím sa minimalizuje riziko vzniku incidentov a straty dát.

2.2 Príklady a porovnania predikcie bezpečnostných udalostí

V oblasti informačnej a kybernetickej bezpečnosti v súčasnosti poznáme mnohé techniky predikcie na zabezpečenie siete. Dnes najpoužívanejšie techniky, môžu byť zoskupené do troch hlavných kategórií, ktorými sú strojové učenie, Markov model a teória Gray [24]. Vybrané techniky predikcií majú rozdielne teoretické a praktické pozadie. Každá z nich má svoje silné vlastnosti aj nedostatky. Ich úplné podrobné opísanie by bolo nad rozsah tejto práce, a teda porovnáme a opíšeme len pár hlavných rozdielov na základe ich opisu v iných štúdiách.

Markov model je matematický model, ktorý popisuje stochastický model, t. j. proces, ktorého výsledok nie je presne predvídateľný, ale je založený na pravdepodobnostných rozdeleniach. Využíva koncept Markovského vlastností, čo znamená, že pravdepodobnosť, s akou bude proces v nasledujúcom kroku vykonaný, záleží len na súčasnom stave procesu a nezávisí od jeho minulých stavov. Je zvyčajne reprezentovaný pomocou stavového diagramu, kde každý stav je znázornený ako uzol a prechody medzi nimi sú zobrazené šípkami. Tieto prechody sú ohodnotené pravdepodobnosťou a určujú pravdepodobnosť prechodu z jedného stavu do druhého [26].

Príkladom použitia Markovho modelu v oblasti bezpečnosti počítačových systémov môže byť určenie celkového rizika zabezpečenia siete. Podrobnejšie sa tomu venujú Nawa Raj Pokhrel a Chris P. Tsokos v článku [34]. Model vyvinutý pre daný článok využíva filtráciu veľkého množstva dostupných informácií stanovením zoznamu priorít zraniteľných uzlov nachádzajúcich sa v sieti. Hĺbkové pochopenie rizík a úrovní priorít každého hostiteľa pomáha jednotlivcom lepšie rozhodnúť napríklad o nasadení bezpečnostných produktov a navrhnúť topológiu siete [34].

Markovove modely dobre fungujú v prítomnosti nepozorovateľných stavov. To umožňuje úspešnú detekciu narušenia a predikciu útoku aj v prípade, že niektoré kroky útoku neboli zistené, resp. ich nebolo možné úplne odvodiť. Existuje niekoľko variantov Markovovho modelu: skrytý Markov model, Markovove modely s premenlivou dĺžkou a Markov model s premenlivým poradím [53].

Skrytý Markov model je často vizualizovaný ako graf. V kybernetickej bezpečnosti útočné triedy sú uzly, pozorovacie symboly sú hrany a pravdepodobnosti sú váhy hrán. V príklade tohto modelu použitého na predikciu útoku sú štyri stavy reprezentujúce postup útočníka z normálneho stavu do stavu pokusu, progresu a úspešného stavu kompromisu. Sendi a kolektív [43] navrhli v roku 2012 metódu prieniku predikcie v reálnom čase, ktorá využíva skrytý Markov model. Viacstupňový útok je hlavným záujmom v danej práci. Experimentálne hodnotenie poukazuje na to, že táto metóda dokáže predpovedať viackrokové útoky, čo je obzvlášť užitočné na zabránenie získania kontroly útočníkom nad stále väčším počtom hostiteľov v sieti [53].

Kholidy s kolektívom publikovali tri články o využití skrytého Markovovho modelu na predikciu útokov. Prvý článok sa venoval detekcii narušenia v cloude [44], potom sa venoval využitiu konečného Markovovho stavu na predikciu viacstupňových útokov v cloude [45] a v poslednom článku sa venovali predikcii narušenia s popísaným konečným kontextom s pravdepodobnostným sufixovým stromom [46].

V tabuľke č. 1 sú zobrazené spomínané modely a ich autori. Je to len zlomok využitia Markovovho modelu pri predikcii v oblasti bezpečnosti na internete. V priebehu času prešiel tento model rôznymi vylepšeniami, čo zefektívňuje prácu pri predikcii. Ciele vylepšenia modelu sú rôzne, no všetky prispievajú k rýchlejšej detekcii anomálii a k lepšej bezpečnosti.

Markov model				
Autor	**Rok**	**Model**	**Evaluácia**	**Cieľ práce**
Pokhrel, Tsokos [34]	2017	Markov chain	CVSS	Celkové riziko zabezpečenia siete
Sendi a kolektív [43]	2012	Hidden Markov model	DARPA 2000	Predikcia ďalšieho kroku pri viackrokovom útoku
Kholidy a kolektív [44], [45], [46]	2014	Hidden Markov model, Variable-order Markov model	DARPA 2000	Metrika načasovania predpovedá útok prichádzajúci o 39 minút

Tabuľka č.1: Príklady Markovovho modelu

Teória Grey (Grey Theory) je matematická metóda na analýzu a predikciu vývoja časových radov. Túto metódu je užitočné využiť pri predikciách s malými

vzorkami a neadekvátnymi informáciami. Využíva sa v rôznych oblastiach, hlavne modely rozšírené v posledných rokoch, ktoré sú viac praktické a predikcie sú presnejšie [25]. V oblasti bezpečnosti na internete má určite aj táto metóda svoje miesto. Dôkazom sú rôzne štúdie poukazujúce na výhody tejto metódy. Môžeme uviesť napríklad model, ktorý je analýzou a výpočtom veľkého množstva informácií získaných zo siete schopný na predikciu aktuálnej bezpečnostnej situácie a jeho budúcu zmenu trendu vytvárať a implementovať relatívnu odozvu podľa výsledkov predikcie. Vďaka tomu je možné znížiť, prípadne vyhnúť sa možným útokom, čím sa zabezpečí chod systému [35].

Ďalším príkladom teórie Grey a jeho rozšírením môže byť príklad práce z roku 2016 od Leau a Manickam [47], kde sa autori snažia prekonať obmedzenia tohto modelu a predstavujú model Grey-Verhulst. Rozšírili pôvodnú diferenciálnu rovnicu tak, aby chyby spôsobené odlišným tvarom od pôvodného časového radu boli čo najviac zredukované. V ďalšej štúdii vylepšili model z tej predchádzajúcej aby použitím Kalmanovho filtra predpovedali ďalšie rezídiá, čím sa zvyšuje presnosť predikcie [48].

Tabuľka č. 2 zobrazuje spomínané modely, kde sa využíva teória Grey. Napriek tomu, že je táto teória menej známa v porovnaní s Markovovým modelom alebo strojovým učením, je taktiež využívaná a má svoje miesto pri predikcii vo všeobecnosti, ale aj pri predikcii v oblasti bezpečnosti.

Grey model				
Autor	**Rok**	**Model**	**Evaluácia**	**Cieľ práce**
Nian a kolektív [35]	2011	Grey model	-	Predikcia aktuálnej bezpečnostnej situácie
Leau, Manickam [47], [48]	2016	Grey-Verhulst model, Grey-Verhulst-Kalman model	DARPA 1999 & 2000	Robustnejšie ako štandardné Grey modely na zvýšenie presnosti predikcie bezpečnostných udalostí

Tabuľka č. 2: Príklady použitia teórie Grey

Treťou spomínanou predikčnou technikou na zabezpečenie siete je **strojové učenie**. Je to oblasť umelej inteligencie zaoberajúca sa algoritmami, ktoré dokážu naučiť počítačový systém riešiť určité úlohy bez toho, aby boli explicitne programované. Hlavnou myšlienkou je, že počítačový systém by mal byť schopný automaticky

identifikovať vzorce a zložité závislosti v dátach a použiť ich na riešenie úloh. Existuje množstvo prác zaoberajúcich sa aplikáciou strojového učenia na predikciu bezpečnosti siete.

Spomeňme napríklad prácu z roku 2012, kde Zheng a kolektív [49] diskutovali o neurónových sieťach so spätným šírením chyby. Zhang a kolektív v roku 2013 [51] porovnávali funkciu spätného šírenia chyby a funkciu radiálnej bázy neurónových sietí, alebo prácu, kde Chen s kolektívom navrhli použitie "small world echo", čo je niečo ako istý druh rekurentného neurónu siete [50]. Neurónové siete boli použité napríklad aj na predikciu narušenia v roku 2016 od Xing-zhu [52].

V tabuľke č. 3 sú zobrazené spomínané články a ciele ich prác. Je to len zlomok prác, no na ukážku, že aj neurónové siete majú zastúpenie v oblasti bezpečnosti na internete, to postačuje. Neurónové siete sú dnes široká téma a ďalej sa rozrastá a napreduje.

Neurónové siete				
Autor	**Rok**	**Model**	**Evaluácia**	**Cieľ práce**
Zheng a kolektív [49]	2012	Backpropagation neural network	KDD99	Modulárny systém, veľmi krátka diskusia
Chen a kolektív [50]	2013	Recurrent neural network	Live (honeypot)	Rekurentná sieť (Small world echo)
Zhang a kolektív [51]	2013	Backpropagation a Radial Basis Function neural network	Vlastné dáta	84,2% - 85,42% presnosť, BP je rýchlejší ako RBF
Xing-zhu [52]	2016	Radial Basis Function neural network	DARPA 1998	Predikcia vniknutia

Tabuľka č. 3: Príklady použitia strojového učenia

3 Neurónové siete

V rámci našej práce sme sa rozhodli využiť na predikciu časových radov neurónové siete. Z už existujúcich štúdií a článkov vieme, že sú dostatočne presné, no stále potrebujú isté rozšírenia, resp. je potrebné hľadať nové prístupy predikcií. To nás viedlo k rozhodnutiu pokúsiť sa o predikciu s ešte nie veľmi známym typom modelu. V tejto kapitole však najprv definujeme neurónové siete vo všeobecnosti a ich základné rozdelenie.

„Neurónová sieť predstavuje navzájom prepojené vrstvy skladajúcich sa z malých jednotiek nazývaných uzly, ktoré vykonávajú matematické operácie na zisťovanie vzorov v dátach“ [60]. Všetky prepojenia medzi neurónmi majú svoje ohodnotenie, čo odborne označujeme ako váhu. Každý neurón patrí do nejakej vrstvy. Záleží to na type modelu, spravidla sa však každá neurónová sieť skladá zo vstupnej vrstvy, z niekoľkých skrytých vrstiev a výstupnej vrstvy. Celé to prerozdelenie neurónov, určenie váh a počet vrstiev nazývame architektúrou siete. Dnes poznáme viacero druhov sietí pre rôzne účely. Neurónové siete sa delia na: dopredná neurónová sieť, rekurentné neurónové siete a konvolučné neurónové siete.

3.1 Dopredná neurónová sieť

Nie veľmi používaná sieť na predikciu časových radov, no patrí k základnému rozdeleniu neurónových sietí, a preto je, myslím si, vhodné popísať aj tento typ. Táto kategória neurónových sietí označuje konkrétny typ architektúry sietí. To znamená, že medzi neurónmi v sieti neexistuje žiadna spätná väzba z výstupov smerom k vstupom. V tomto prípade poznáme jednovrstvovú doprednú neurónovú sieť alebo viacvrstvovú.

Poznáme avšak **algoritmus spätného šírenia chyby**. Najčastejšie používaný na tréning doprednej neurónovej siete. Spätne aktualizuje synaptické váhy sietí šírením gradientového vektora, v ktorom je definovaný každý prvok ako derivácia chybového rozsahu vzhľadom na parameter. Rozdielom medzi aktuálnymi sieťovými výstupmi a požadovanými výstupmi sa definujú chybové signály [54].

3.2 Rekurentné neurónové siete

Rekurentné neurónové siete majú schopnosť dynamicky kombinovať skúsenosti vďaka ich vnútornému opakovaniu. To znamená, že vykonáva rovnakú funkciu pre každý vstupný x_t a výstupný y_t, ktorý je závislý od vstupného x_t, ale aj predchádzajúceho stavu h_{t-1}.

Teda stav v skrytej vrstve v čase t vieme vypočítať ako:

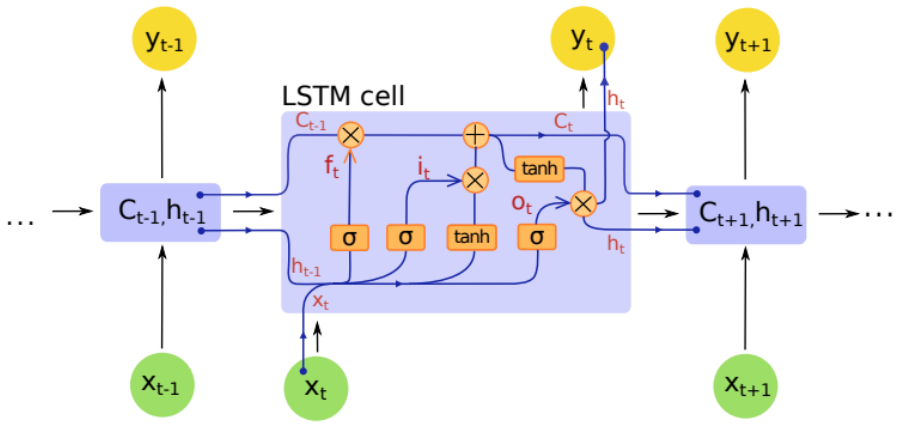
$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b_h)$$

Výstup siete v čase t počítame pomocou nasledujúceho vzorca:

$$y_t = \sigma(W_y h_t + b_y),$$

kde x_t je vstupný vektor, h_t je stav skrytej vrstvy a y_t je výstupný vektor. W a b sú učiace parametre a σ je aktivačná funkcia [12].

Tieto výpočty sa opakujú pre každý krok x_t sekvencie [20].



Obr. 4: Model LSTM a jeho jednotka. Prevzaté z [12]

Na obrázku č. 4 je zobrazená LSTM sieť. Hlavnou výhodou tejto neurónovej siete je, že nemá stanovenú veľkosť vstupu. Táto schopnosť je kľúčová a vyplýva z vlastnosti siete, že všetky váhy neurónov v sieti sú zdieľané každej časti sekvencie. Okrem prepojenia na ďalšiu vrstvu obsahujú aj rekurzívne prepojenie na seba. To umožňuje čiastočne si pamätať, čo bolo predtým na vstupe [20].

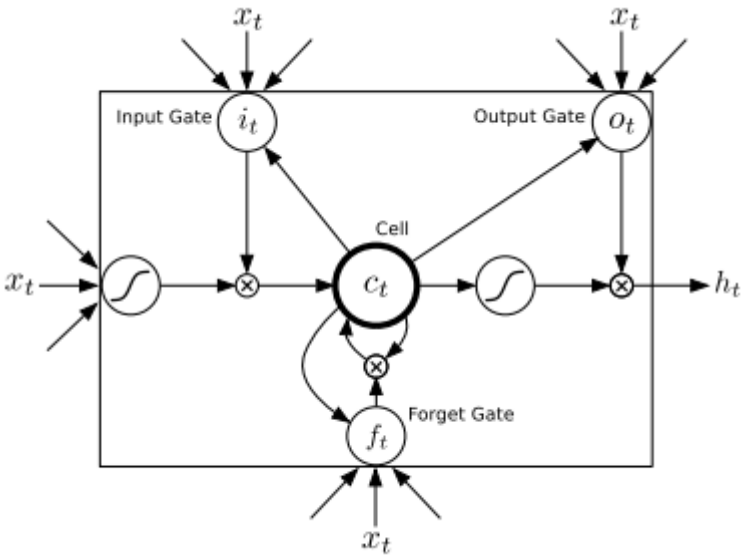
3.2.1 Model LSTM

Jednoduché rekurentné siete však obsahujú aj isté nedostatky. Majú internú pamäť na spracovanie sekvenčných údajov a nastáva tam problém miznúceho

gradientu.Sieť LSTM je trénovaná pomocou spätného šírenia chyby v čase a prekonáva problém miznúceho gradientu [14]. Takisto prináša lepší výkon pri spracovaní dlhých sekvencií, na rozdiel od jednoduchých rekurentných sietí, kde vznikajú problémy [12].

Siete LSTM majú namiesto neurónov pamäťové bloky, ktoré sú prepojené cez vrstvy [14]. Každý blok obsahuje pamäťové bunky s vlastným spojením, ktoré sú schopné zapamätať si dočasný stav siete a špeciálne multiplikatívne jednotky nazývané brány [16]. Všetky bloky obsahujú tiež brány, ktoré riadia stav a výstup [14]. Konkrétne ide o vstupnú bránu (Input Gate), výstupnú bránu (Output Gate) a pamäťovú bránu (Forget Gate) [16].

Každá má svoju charakteristickú úlohu: **pamäťová brána** – podmienečne rozhoduje zapamätaný skrytý stav, **vstupná brána** – spracováva hodnoty zo vstupu a rozhoduje, ktoré z nich majú aktualizovať stav pamäte, **výstupná brána** – určuje, čo sa má vydať na základe vstupu a pamäte bloku [14].



Obr. 5: Samostatná pamäťová jednotka v LSTM modeli. Prevzaté z[23]

Pre porovnanie na obrázku č. 6 zobrazujeme výpočet skrytého stavu, označeného ako h_t, v sieti LSTM. Už na prvý pohľad vidieť vyššiu zložitosť a dôslednosť pri výpočte. Celkový výsledok skrytého stavu ovplyvňujú už spomínané brány, teda i_t je vstupná brána, f_t predstavuje zabudnutú bránu a o_t je výstupná brána. Medzi samotnými bránami sú váhy, ktoré ovplyvňujú výsledok.

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

Obr. 6: Výpočet skrytého stavu v sieti LSTM. Prevzaté z [20]

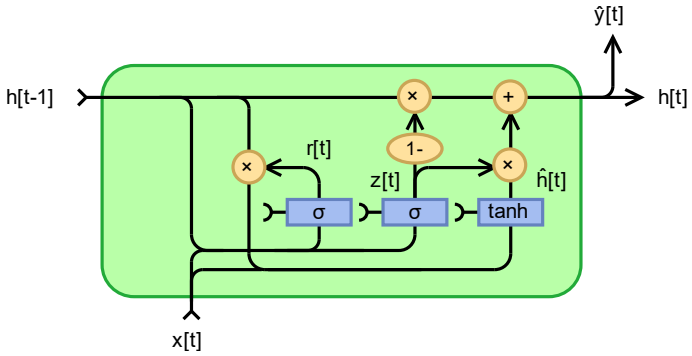
V súčasnosti poznáme viacero typov modelu LSTM, ktoré môžu byť použité pre rôzne špecifické predikcie časových radov. Predikcia časových radov môže byť jednorozmerná, viacrozmerná, viacstupňová alebo aj viacrozmerná viacstupňová [13].

Ako sme už spomínali, existuje mnoho rôznych typov modelov, ktoré využívajú architektúru LSTM. Z tých, ktoré vytvárajú predikciu pre jednorozmerný vstup, poznáme napríklad „Vanilla LSTM “, „Stacked LSTM“, „obojsmerný LSTM“, alebo aj hybridné modely ako „CNN-LSTM“ či „Conv-LSTM“ [15].

3.2.2 Model GRU

“Gated recurrent units” je ďalším typom rekurentnej siete, ktorý je veľmi podobný modelu LSTM. Medzi hlavné rozdiely medzi modelom GRU a modelom LSTM patrí menší počet brán. GRU taktiež nemá žiadnu vnútornú pamäť. GRU je práve preto podstatne rýchlejší model s menším obnosom dát na trénovanie. Avšak ak je k dispozícii dostatočne veľký počet dát a zakladáme si na presnosti, tak model LSTM dosahuje lepšie výsledky.

Model GRU obsahuje dve brány: **aktualizačnú bránu** (Update gate)– zodpovedá za určenie množstva informácií, ktoré sa odovzdávajú v ďalšom stave, **zabúdacia bránu** (Reset gate) – rozhoduje o tom, koľko z minulých informácií je potrebné zanedbať. Najprv zabúdacia brána ukladá informácie z minulého časového kroku do nového obsahu pamäte. Vstupný vektor sa vynásobí so skrytým stavom ich váh. Po prvkoch sa vynásobí zabúdacia brána a predtým skrytý násobok stavu. Sčítaním predchádzajúcich výsledkov sa aplikuje nelineárna aktivačná funkcia a generuje sa ďalšia sekvencia [55].



Obr.7: Štruktúra GRU [56]

3.3 Konvolučné neurónové siete

Konvolučné neurónové siete (Convolution Neural Networks), označované ako CNN, sú založene na doprednej propagácii využívajúcej viacero skrytých vrstiev [17]. Táto sieť má svoje charakteristické vlastnosti, ktoré v určitých prípadoch tvoria veľkú výhodu oproti klasickým dopredným sieťam. Prvou základnou vlastnosťou je lokálne prepojenie. To znamená, že umožňuje neurónu v skrytej vrstve napojenie na oblasť neurónov v predchádzajúcej vrstve (anglicky receptive field). Uvedená vlastnosť je prebraná z vizuálnej časti mozgu. Ďalšou vlastnosťou konvolučných sietí je n-dimenzionálny rozmer vrstiev [20]. Napríklad na vstupnej vrstve má sieť tri dimenzie, a to výšku, šírku a hĺbku [21]. Treťou, no už kritickou vlastnosťou je, že určité váhy sú zdieľané, čiže všetky neuróny v konvolučnej vrstve v určitej hĺbke majú zdieľané váhy. Táto sieť nachádza najlepšie využitie v prípade, keď máme na vstupnej vrstve pre obrázok mnoho neurónov a obrovský počet prepojení váh na ďalšiu vrstvu [20].

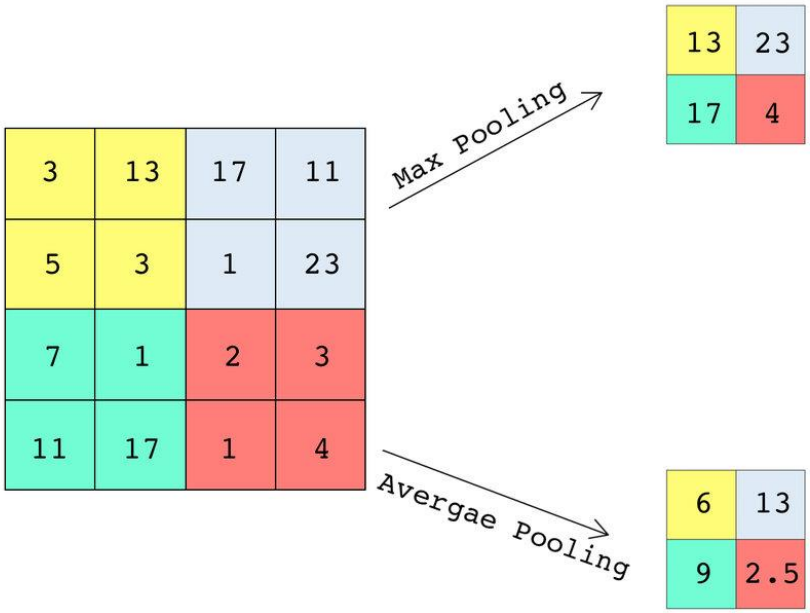
Vrstvy môžu byť **konvolučné, plneprepojené vrstvy, pooling vrstva, softmax** a **vrstva obsahujúca ReLu neuróny**. Vhodnou kombináciou týchto vrstiev vznikne umelá neurónová, ktorá vyniká pri spracovávaní obrazu [18].

Parametre v **konvolučnej vrstve** pozostávajú zo sady naučiteľných filtrov. Každý z nich je relatívne malý, no siaha cez celú hĺbku vstupného objemu. Keď posúvame filter po šírke a výške vstupného objemu, zakaždým vytvoríme dvojrozmernú aktivačnú mapu, ktorá nám poskytuje odozvu tohto filtra v každej priestorovej polohe. Sieť sa intuitívne naučí filtre, ktoré sa aktivujú, keď uvidia istý typ vizuálneho prvku. Môže to byť orientácia alebo nejaká farba na prvej vrstve. Tieto aktivačné mapy sa naskladajú pozdĺž hĺbkovej dimenzie a takto vznikne výstupný objem. Hyperparameter,

určujúci priestorový rozsah, nazývaný receptívne pole neurónu, je veľkosť filtra. Rozsah konektivity pozdĺž osi hĺbky sa vždy rovná hĺbke vstupného objemu.

Veľkosť výstupného objemu závisí od ďalších hyperparametrov, a to hĺbky (depth), kroku (stride) a nulovej vložky (zero-padding). Hĺbka zodpovedá počtu filtrov, ktoré by sme chceli použiť. Krok je veľkosť, ktorá určuje, koľko pixelov naraz chceme preskočiť, keď posúvame filter. Nulová vložka nám umožňuje lepšiu manipuláciu priestorovej veľkosti výstupných objemov. Hlavnou vlastnosťou tohto hyperparametra je, že si vieme zachovať priestorovú veľkosť vstupnej a výstupnej šírky a výšky [21].

Funkciou **pooling vrstvy** je zmenšovať priestorovú veľkosť, čím sa zníži aj množstvo parametrov a výpočtov v sieti, a teda slúži aj ako kontrola pre nadmerné vybavovania [21]. Môžeme si to predstaviť ako zovšeobecnenie konkrétnej vzorky. To v konečnom dôsledku urýchľuje aj tréning siete. Táto vrstva si neukladá žiadne váhy, slúži výhradne na transformáciu údajov. Poznáme viacero druhov pooling vrstvy, napríklad MaxPooling, ktorá vyberá maximálnu hodnotu, alebo AvgPooling, kde vrstva počíta priemer vstupnej oblasti. Na obrázku nižšie zobrazujeme vstup a výstup oboch typov týchto vrstiev.



Obr. 8: Vstup a výstup vrstiev „Max Pooling“ a „Avg Pooling“ [33]

Vrstva **softmax** vytvára pravdepodobnosti pre každý možný výstup, čím umožňuje rýchlejšiu konvergenciu počas tréningu [22]. Aktivačnú vrstvu v tejto sieti predstavuje **ReLU vrstva** (Rectified Linear Unit). Môže sa napájať priamo na

konvolučné vrstvy, plneprepojené alebo aj na pooling vrstvy [20]. V tejto vrstve sa odstraňujú záporné hodnoty z filtrovaného vstupu. Teda ak je vstup pod nulou, výstupom bude nula. Ak je však vstup väčší ako nula, bude mať lineárny vzťah s hodnotou príslušnej vstupnej premennej. To opäť v konečnom dôsledku urýchľuje samotný tréning siete.

Najčastejšie sa teda tento model siete využíva na rozpoznávanie obrázkov alebo videoanalýzy. Konvolučné neurónové siete však vieme využiť aj na predikciu časových radov. Poznáme viacero možných druhov tohto modelu na predikciu: jednorozmerný model, viacrozmerný, viacstupňový a viacrozmerný viacstupňový CNN model.

Pretože chceme porovnávať rôzne druhy sietí pre jednorozmerný vstup, CNN model pre jednorozmerný vstup by sa mohol skladať z konvolučnej skrytej vrstvy, v istých prípadoch by mohla nasledovať druhá konvolučná vrstva. Príkladom pre potrebu použitia druhej konvolučnej vrstvy by mohla byť dlhá vstupná frekvencia. Sploštená vrstva síce nebola popísaná v hlavnom opise konvolučnej sieti, no mohla by byť vhodným doplnkom medzi konvolučnou časťou sieti a plneprepojenou vrstvou, na zmenšenie mapy na jednorozmerný vektor. Plneprepojená vrstva tak interpretuje vlastnosti extrahované z konvolučnej časti modelu [19].

4 Dáta

Dáta vo všeobecnosti predstavujú jedinečný súbor údajov zozbieraných zo systémov na detekciu narušenia. Údaje umožňujú lepšie pochopenie súčasných útokov a správanie sa protivníkov z viacerých uhlov pohľadu. Slúžia aj pri vyhodnotení a porovnaní ich prístupov, a to na analýzu výstrah a detekciu narušenia. Ponúkajú nám aktuálne bezpečnostné výstrahy a tým odrážajú súčasnú situáciu v oblasti informačnej a kybernetickej bezpečnosti [9].

Zozbierané dáta nám umožňujú experimentovanie s pokročilými metódami agregácie výstrah alebo aj rekonštrukciu scenára útoku a projekciu útoku. To umožňuje odvodiť pohľady na správanie sa útočníkov, čím vieme lepšie prispôsobiť detekciu a projektovanie nových mechanizmov na reakcie na incidenty, ktoré sú účinné proti súčasným kybernetickým hrozbám [9].

4.1 Warden

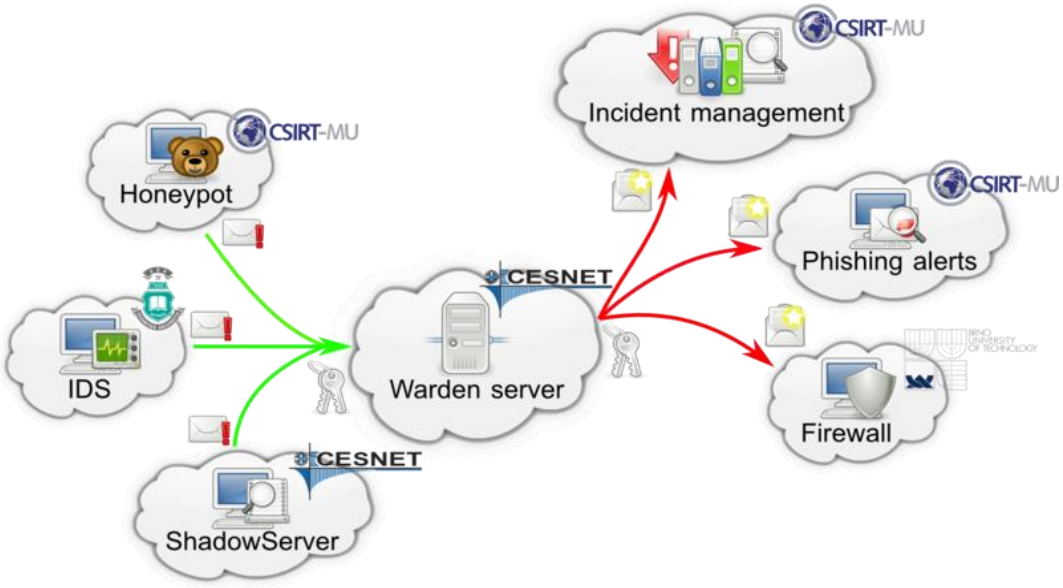
V našej práci spracovávame bezpečnostné udalosti z platformy Warden. Je to systém na rýchle a efektívne zdieľanie a využitie informácií o zistených hrozbách. Nazbierané dáta sú výsledkom nasadených nástrojov do ich monitorovacích sietí. To umožňuje a uľahčuje ďalším bezpečnostným tímom prácu pri monitorovaní a zlepšovaní bezpečnosti v sieti [10].

Cieľom projektu Warden je najmä vytvorenie jednoduchého, robustného a bezpečnostného systému na zvýšenie bezpečnosti v sieti, na prehĺbenie spolupráce, komunikáciu či zdieľanie informácií medzi jednotlivými tímami a organizáciami vo svete [10].

Systém sa skladá z hlavného Warden serveru, ktorý obstaráva uvedenú zásadnú činnosť, a dvoch typov klientov. Prvým typom je odosielajúci klient, ktorý sa stará o dodávanie informácií poskytovaných zapojenými organizáciami na Warden server. Druhý typ predstavuje odoberajúceho klienta, určeného na získavanie informácií požadovaných zapojenou organizáciou. Serverová strana systému zaisťuje prijatie a ukladanie informácií zasielaných klientami. Poskytuje prístup ku všetkým uloženým udalostiam na serveri, ktoré sú chránené autentizáciou [10].

Samotná udalosť (event) v systéme Warden predstavuje informáciu o zdroji bezpečnostnej hrozby, ktorá bola zaznamenaná niektorým zo zapojených členov. Tieto

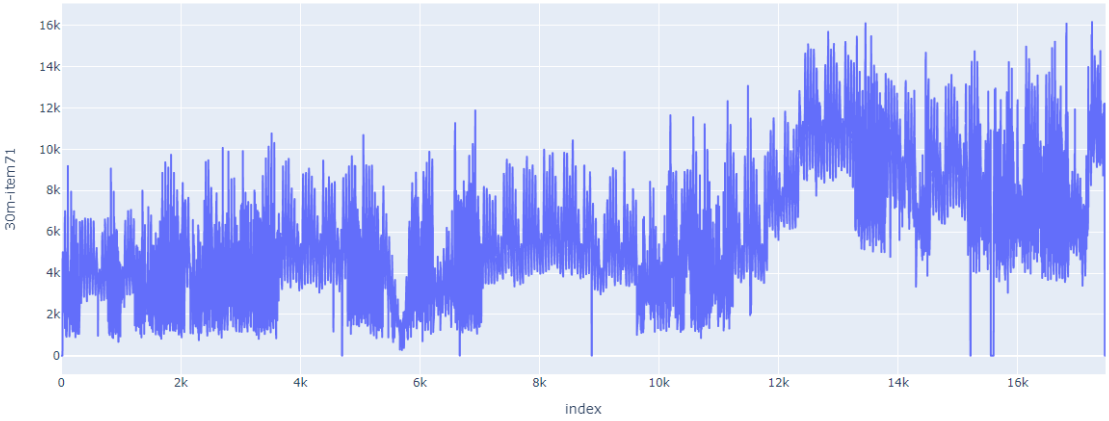
informácie môžu pochádzať z rôznych zdrojov, ide napríklad o detekčné systémy spolupracujúcej organizácie, ako sú IDS, sieťové pasce (honeypoty) alebo sú to dáta tretích strán (Honeynet, Shadowserver…) [10]. Na obrázku 9 zobrazujeme samotnú architektúru projektu Warden so vstupmi a výstupmi.



Obr. 9: Warden architektúra, Prevzaté z [10]

4.2 Vybraný dataset

Spracovávame dáta s veľkosťou 1 stĺpec a vyše 17 000 riadkov bezpečnostných udalostí. Pôvodné dáta boli rozsiahlejšie, no vzhľadom na našu prácu budú postačovať aj takto zmenšené dáta. Vybrali sme si konkrétne stĺpec s informáciou o počte prístupov na port 445/TCP. Tento port poznáme aj pod označením smb, resp. Samba. Ide o protokol, ktorý vo Windows umožňuje komunikáciu medzi aplikáciami a službami na počítačoch. Podporuje základné sieťové služby, ako je tlač, zdieľanie súborov a zariadení [11].



Obr. 10: Časový rad zobrazujúci počet prístupov na port 445/TCP

Na obrázku 10 zobrazujeme náš časový rad vstupných dát. Pri práci sme sa museli vyrovnať s chýbajúcimi údajmi, čo sme vyriešili obkročením, to znamená spojením údajov pred medzerou a za medzerou. Rozdelili sme dáta na trénovaciu množinu o veľkosti 16 522 hodnôt a validačnú množinu o veľkosti 1 241 hodnôt. Sú to dáta zbierané v 30-minútovom intervale. Jeden bod na tomto grafe predstavuje počet pokusov o prístup na port 445/TCP za posledných 30 minút. Na vytvorenie jednej predikcie sme použili dáta nazbierané počas ôsmich dní, čo zodpovedá 384 predchádzajúcim hodnotám.

5 Model Transformer

Pre našu prácu sme si vybrali už spomenutý model Transformer, ktorý sa dostáva do popredia v rôznych oblastiach, napríklad aj pri predikcii alebo preklade viet. Napriek tomu, že je to nový model publikovaný prvýkrát v roku 2017 tímom Google Brain, dosahuje lepšie výsledky ako klasické rekurentné alebo konvolučné neurónové siete. Táto architektúra odstraňuje potrebu používať rekurentné neurónové siete implementáciou mechanizmov pozornosti (attention) a sebapozorovania (self-attention) [4].

5.1 Kóder

Skladá sa zo **vstupnej vrstvy**, **pozičnej kódovacej vrstvy** a **sady štyroch identických kóderov**. **Vstupná vrstva** mapuje údaje zo vstupných časových radov na vektor dimenzie modelu cez plne prepojenú sieť. Aby mohol model využívať mechanizmus pozornosti s viacerými hlavami, tieto kroky sú nevyhnutné. **Pozičné kódovanie** so sínusovými a kosínusovými funkciami sa používa na kódovanie sekvenčných údajov z časových radov pridávaných postupne po jednotlivých prvkoch. Pridávanými údajmi teda sú: vstupný vektor a vektor pozičného kódovania. Výsledný vektor sa privádza do **štyroch vrstiev kódera**. Každá vrstva kódera pozostáva z ďalších dvoch podvrstiev: samoupozorňovacej podvrstvy a plne prepojenej doprednej podvrstvy. Každá podvrstva nasleduje do normalizačnej vrstvy. Kóder takýmto spôsobom vytvára vektor v potrebnom rozmery, ktorý privádza na vstup do dekodéra [12].

5.2 Dekóder

Dekodér sa skladá zo **vstupnej vrstvy**, **štyroch identických vrstiev dekodéra** a **výstupnej vrstvy**. **Vstup** dekodéra začína posledným dátovým bodom zo vstupu kódera. Vstupná vrstva, podobne ako aj pri kóderi, mapuje vstup dekodéra na vektor dimenzie modelu. Navyše k dvom podvrstvám v každom kóderi vkladá dekodér tretiu podvrstvu na uplatnenie mechanizmu sebapozornosti nad výstupom kódera. Nakoniec je tam **výstupná vrstva**, ktorá mapuje výstup poslednej vrstvy dekodéra do cieľovej časovej sekvencie. Správne nastavenie dekódera zabezpečuje, že predikcia jednotlivého bodu v časovom rade bude závisieť iba od predchádzajúcich bodov dát v časovom rade [12].

5.3 Mechanizmus pozornosti

Prvotne bol mechanizmus navrhnutý na riešenie problémov pri počítačovom videní, ako je napr. Klasifikácia obrázkov. Neskôr, využitím hybridných modelov, kombinujúc mechanizmus pozornosti a rekurentnej neurónovej siete, dosiahli výborný výkon pri predikcii časových radov, čo ich viedlo k hlbšiemu výskumu a výsledkom je už spomenutý komplexný model Transformer [7].

Tento mechanizmus vieme charakterizovať rovnicou:

$$C(Q, K, V) = softmax(f(Q, K)) * V,$$

kde *Q, K, V* znamená: dopyt (query), kľúč (key) a hodnota (value). Tento systém na vyhľadávanie informácií zaviedol Vaswani a spol. Vo svojej práci [3]. Je to analógia, kde sa dopyt používa na vyhľadanie zodpovedajúceho kľúča (alebo najpodobnejšieho) a získanie jeho hodnoty [5]. Teda vo všeobecnosti je jadrom tohto mechanizmu výpočet relatívnej váhy medzi kóderom a dekóderom [7].

Podobne ako architektúry seq2seq sú transformátory schopné mapovať vstupnú sekvenciu na výstupnú sekvenciu s potenciálne rôznymi dĺžkami. Podobne sa skladajú aj z dvoch blokov: kódera a dekódera [5].

5.4 Mechanizmus sebapozorovania

Pri mechanizme pozornosti kladieme dôraz na použitie výpočtov v cieľovom prvku. Tam sa zdroj a cieľ líšia. V tomto mechanizme sebapozorovania sa cieľ rovná zdroju [7]. Inými slovami, v sebapozorovaní sa mechanizmus pozornosti deje vnútri zdroja alebo cieľa. Teda výpočet je podobný, no predmety výpočtu sú odlišné.

Táto modelová architektúra sa vyhýba opakovaniu výpočtov a spolieha sa výlučne na mechanizmus pozornosti a na modelovanie závislosti medzi vstupom a výstupom. Využíva viac paralelizácie [8], čím predovšetkým umožňuje trénovanie na väčších súboroch údajov v podstatne kratšej dobe a dosahuje vďaka tomu kvalitnejšie výsledky.

Na trénovanie sa používa technika „vynucovania učiteľa“. Dôvodom je, aby bolo možné trénovanie vykonať v jednom výpočte na dávku [5]. Teda priamo súvisí s rýchlosťou trénovania. V iných modeloch a v iných zdrojoch literatúry, napríklad

Vaswani (2017), sa taktiež uvádzajú dobré výsledky modelov, ktoré preferujú túto techniku. Keďže využíva napájanie dekódera s cieľovou sekvenciou posunutou doprava o jednu vzorku, na výpočet ďalších krokov sa musí použiť automatická regresia [5]. Budúce kroky časových radov teda nie sú známe, no predpovedanú vzorku vieme vrátiť týmto spôsobom späť na vstup, kde sa použije na predpovedanie ďalšej vzorky.

6 Výsledky

6.1 Použité technológie

Vytvorili sme model neurónovej siete typu Transformer určený na predikciu dát. Základ siete spočíva v modeli predstavenom v diele „Attention is all you need“ od A. Vaswani a spol. Využili sme prostredie **Colaboratory**, tiež známe ako Google Colab, ktorý je vlastnený a vyvíjaný spoločnosťou Google. Je to cloudová platforma, poskytujúca prostredie Jupyter notebook na písanie a spúšťanie Python kódu. Google Colab je bezplatná služba, ktorá umožňuje používateľom písať a spúšťať kód, spolupracovať s inými používateľmi a mať prístup k rôznym knižniciam a frameworkom pre strojové učenie, bez potreby inštalácie na lokálnom počítači alebo výkonného hardvéru.

Využili sme aj otvorenú softvérovú knižnicu **TensorFlow**. Bola vyvinutá spoločnosťou Google Brain a je jedným z najpoužívanejších nástrojov pre tvorbu a nasadenie rôznych typov modelov strojového učenia. Poskytuje širokú škálu nástrojov, knižníc a rozhraní, ktoré umožňujú vytvárať a trénovať modely určené a rôzne úlohy v oblasti umelej inteligencie. Je dostupný pre viaceré programovacie jazyky, vrátane Pythonu, C++ a iné. Poskytuje aj vysoko úrovňové API (z angl. Application Programming Interface), známe ako Keras.

Túto dostupnú otvorenú knižnicu sme taktiež využili v našej práci, keďže spomínaný **Keras** sa vyznačuje jednoduchosťou a flexibilitou pri definovaní architektúry modelu. Je typický aj minimálnym množstvom kódu, čo v podstate zjednodušuje celý proces vytvárania a trénovania neurónových sietí.

Ďalšou otvorenou knižnicou pre programovanie v jazyku Python, ktorú sme využili, je knižnica **Pandas**. Tá je navrhnutá na efektívne spracovanie a manipuláciu tabuľkových dát. Je jednou z najpoužívanejších knižníc pre analýzu a spracovanie dát, a to aj vďaka rôznym iným nástrojom, ktoré ponúka, ako napríklad čistenie a transformáciu dát, spracovanie chýbajúcich hodnôt, manipuláciu s dátumami a časmi, kategorizáciu dát a mnoho ďalších.

Výkonné nástroje pre výpočty s viacrozmernými poliami a maticami poskytuje otvorená knižnica **NumPy** (z angl. Numeric Python). Umožňuje efektívne ukladanie a manipuláciu s veľkými objemami dát. Je rýchly a efektívny práve pre operácie s vektormi, ktoré umožňujú vykonávať matematické operácie na celých poliach naraz, bez

potreby iterovania cez jednotlivé prvky. Obsahuje aj nástroje napríklad na generovanie náhodných čísel, triedenie, zlučovanie alebo transformáciu polí.

Na vizualizáciu dát sme použili knižnicu **Plotly**, ktorá je k dispozícii v niekoľkých programovacích jazykoch vrátane Pythonu. Umožňuje vytvárať rôznorodé vizualizácie, ako sú napríklad grafy, tabuľky, mapy a mnoho ďalších. Vyznačuje sa aj interaktivitou a dynamikou. Jeho vizualizácie umožňujú používateľom preskúmať a analyzovať dáta pomocou zobrazovania alebo zámerného skrývania určitých častí grafu, dokáže prispôsobiť rozsah osí a iné. Z tejto knižnice sme importovali aj niektoré moduly, konkrétne „graph_objects“, „subplots“, „io“ a „express“. Všetky tieto moduly knižnice Plotly vo všeobecnosti obsahujú ďalšie triedy uľahčujúce prácu s dátami a ich vizualizáciou.

Poslednou využitou knižnicou je **Pickle.** Tá umožňuje konverziu (serializáciu) objektov do binárneho formátu a následné obnovenie (deserializáciu) týchto objektov späť do pamäte. Má aj pár obmedzení, napríklad že je špecifická iba pre jazyk Python, čo ale v našom prípade postačuje.

6.2 Predikcia: premenné a hodnoty

V práci sme sa venovali predikcii jednokrokovej aj viackrokovej. Konkrétne sme trénovali na 10-krokovom a aj 20-krokovom modeli. Rozdiel medzi jednokrokovou a viackrokovou predikciou je, že zatiaľ čo pri jednokrokovej predikcii model predpovedá jednu budúcu hodnotu, viackroková predikcia predpovedá postupnosť budúcich hodnôt určitého rozsahu [36], teda v našom prípade desať a dvadsať budúcich hodnôt, ktoré dáva ako výstup modelu.

Naprogramovaný model bol spúšťaný na vybraných dátach, no opakovane s rôznymi parametrami. Prvý tréning spočíval na desiatich epochách. Menili sme hodnoty „**head size**“, „**num heads**“, „**ff dim**“ a „**num transformer blocks**“.

Parameter „**head size**“ určuje veľkosť vstupno-výstupnej vrstvy pre každú hlavičku v sieti (attention head). Čím je hodnota „head size“ väčšia, tým viac informácií môže byť zachytených v každej hlavičke, no zároveň sa zvyšuje celkový počet parametrov modelu a teda aj jeho náročnosť na výpočet a trénovanie. My sme sa rozhodli trénovať model s hodnotami tohto parametra 256 a 128.

Ďalším parametrom je „**num heads**“, čo je počet hlavičiek (attention head) v sieti. Umožňuje rozdeliť vstupnú sekvenciu na viacero menších dimenzií. Každá hlavička predstavuje jeden smer pozornosti, takže siete Transformer môžu extrahovať rôzne vlastnosti vstupu paralelne v rôznych častiach siete [28]. Znovu platí, že čím je vyšší počet hláv v sieti, tým je viac rôznych vzťahov medzi rôznymi časťami vstupu a tým sa zvyšuje aj jeho náročnosť na výpočet a trénovanie. Náš model sme trénovali s hodnotami 2 a 4.

Dopredná vrstva („feed-forward layer“) prehlbuje sieť a využíva lineárne vrstvy na analýzu vzorov vo výstupe vrstiev pozornosti [29]. Táto vrstva sa nachádza v každom kóderi a dekóderi [3]. Veľkosť „**ff dim**“ určuje veľkosť skrytej vrstvy v doprednej sieti vnútri transformátora [30]. Tréning sme spúšťali na hodnotách 2 a 4.

„**Num transformer blocks**“ udáva počet blokov (vrstiev) transformera. Jedna vrstva vydáva jeden vektor pre každý časový krok z našej vstupnej sekvencie. Tu berieme priemer naprieč všetkými časovými krokmi a na klasifikáciu sa používa dopredná sieť [30]. Opäť sme tu menili hodnoty 2 a 4.

Vyskúšaním všetkých možných kombinácií sme dosiahli 16 natrénovaných modelov siete Transformer. Sledovali sme hodnoty „**loss**“ a „**val_loss**“, ktoré sme následne spracovali do tabuľky a poznačili sme, po ktorej iterácii je hodnota val_loss väčšia ako hodnota loss. V našej práci používame metriku „mae“.

Hodnota „**loss**“ (strata) predstavuje chybu alebo odchýlku medzi predikovanou hodnotou a skutočnou hodnotou pre konkrétnu úlohu strojového učenia. Používa sa na monitorovanie tréningu modelu a vyhodnocovanie jeho výkonu. Po skončení každej epochy sa vypočíta hodnota loss a táto hodnota sa používa na aktualizáciu parametrov modelu pomocou algoritmu spätného šírenia chyby („backpropagation“). Cieľom je minimalizovať hodnotu loss pre validačné dáta, aby sa dosiahol čo najlepší výkon modelu.

Priemerná absolútna chyba („**MAE**“ z angl. Mean Absolute Error) je jednou z mnohých metrík na zhrnutie a posúdenie kvality modelu strojového učenia. Vypočítava sa odčítaním predpokladanej hodnoty od skutočnej hodnoty. Táto chyba predikcie sa vyskytuje pri každom zázname a z každej chyby sa vezme absolútna hodnota. Z toho sa vypočíta priemer pre všetky zaznamenané absolútne chyby. Teda je to priemerný súčet všetkých absolútnych chýb [32].

Stratová funkcia na validačnej množine („**val loss**“) sa používa na hodnotenie výkonnosti modelu hlbokého učenia na validačnej sade dát. Validačná množina je časť údajov vyčlenená na overenie výkonnosti modelu. Stratová funkcia na validačnej množine je podobná stratovej funkcii na trénovacej množine a vypočíta sa zo súčtu chýb pre každý príklad v sade validácie. Okrem toho sa po každej epoche meria strata validácie. Informuje nás o tom, či model potrebuje ďalšie ladenie (prípadne úpravy) alebo nie [31]. Cieľom je minimalizovať hodnotu val loss funkcie pre validačnú sadu, aby sme dosiahli čo najlepší výkon modelu.

„**Val MAE**“ (validation mean absolute error) je metrika používaná na vyhodnocovanie presnosti modelu pri regresných úlohách počas validačnej fázy trénovania. Teda podobne ako MAE, val MAE vyjadruje priemernú absolútnu odchýlku medzi predikovanou a skutočnou hodnotou, ale je vypočítaná na validačnej sade. Cieľom je minimalizovať hodnotu Val MAE, čo znamená minimalizovať odchýlku medzi predikciou modelu a skutočnými hodnotami na validačných dátach. Nižšia hodnota Val MAE indikuje lepšiu presnosť modelu na validačných dátach. Val MAE je dôležitým ukazovateľom pri vyhodnocovaní modelu počas trénovania, aby sme mohli sledovať jeho výkon a prípadne optimalizovať model alebo upraviť jeho parametre.

6.3 Výsledky

V kapitole 4.1 sme opísali výber dát a obmedzenia, s ktorými sme sa museli vysporiadať. Tieto dáta sme museli normalizovať pred tréningom. To znamená, že hodnoty zo súboru sa transformovali do škály (0,1) a od týchto hodnôt sme odrátali hodnotu „train_mean“ a vydelili hodnotou „train_std“. Presne opačný spôsob sme využili neskôr pre denormalizáciu dát, teda výsledky z predikcie sme vynásobili hodnotou „train_std“ a následne pripočítali hodnotu „train_mean“.

Modely s najlepšími hodnotami sme vybrali a trénovali dlhšie. Čo sa týka jednokrokovej predikcie, najlepšie výsledky poukazujú na najmenší model, čiže model s hodnotami head_size = 128, num_heads = 2, ff_dim = 2, num_transformer_blocks = 2. V tabuľke č. 4 zobrazujeme výsledky tejto siete v počiatočnom tréningu na desiatich epochách. Hodnota val_mae je najnižšia zo všetkých sietí. Hodnota mae (loss) je o niečo vyššia ako v iných sieťach, no najdôležitejšou je v tomto prípade hodnota val_mae.

V tabuľke č. 5 zobrazujeme hodnoty výpočtov po desiatej epoche všetkých trénovaných sietí.

128;2;2;2

Epoch	loss	mae	val_loss	val_mae
1	0.8280	0.8280	0.5908	0.5908
2	0.6222	0.6222	0.5777	0.5777
3	0.5511	0.5511	0.5748	0.5748
4	0.5150	0.5150	0.5825	0.5825
5	0.4861	0.4861	0.5821	0.5821
6	0.4635	0.4635	0.5560	0.5560
7	0.4490	0.4490	0.5432	0.5432
8	0.4367	0.4367	0.5295	0.5295
9	0.4275	0.4275	0.5404	0.5404
10	0.4187	0.4187	0.5108	0.5108

Tabuľka č. 4: Hodnoty modelu Transformer s hodnotami head_size = 128, num_heads = 2, ff_dim = 2, num_transformer_blocks = 2 (jednokroková predikcia)

	mae	Val_mae			mae	Val_mae
256,2,2,2	0.4255	0.5644		**128,2,2,2**	0.4187	0.5108
256,2,2,4	0.4328	0.5441		**128,2,2,4**	0.4333	0.5627
256,2,4,2	0.4244	0.5251		**128,2,4,2**	0.4158	0.5611
256,2,4,4	0.4223	0.5596		**128,2,4,4**	0.4363	0.5278
256,4,2,2	0.4303	0.5695		**128,4,2,2**	0.4181	0.5325
256,4,2,4	0.4420	0.5487		**128,4,2,4**	0.4334	0.5343
256,4,4,2	0.4336	0.5385		**128,4,4,2**	0.4310	0.5581
256,4,4,4	0.4243	0.5430		**128,4,4,4**	0.4359	0.5830

Tabuľka č. 5: Hodnoty modelov Transformer po počiatočnom tréningu (10 epoch) – jednokroková predikcia

Vo všeobecnosti platí, že čím je model menší, jednoduchší, aj výpočty sú menej zložité. Zaujímavé sú však hodnoty ďalších sietí, kde už neplatí toto všeobecné pravidlo. Farebne sme rozlíšili modely s najnižšími hodnotami v tabuľke č. 5. Najtmavšie zvýraznené sú hodnoty najlepšie, teda najnižšie. Čím svetlejšie je zvýraznený model, tým je vyššia hodnota, no stále sa zaraďuje medzi najlepšie. Vybrali sme štyri typy siete

s najlepšími hodnotami val_mae. Vychádzajú nám teda siete zvýraznené žltou farbou, t. j. s hodnotami

head_size=256, num_heads=2, ff_dim=4, num_transformer_blocks=2;

head_size=128, num_heads=2, ff_dim=2, num_transformer_blocks=2;

head_size=128, num_heads=2, ff_dim=4, num_transformer_blocks=4;

head_size=128, num_heads=4, ff_dim=2, num_transformer_blocks=2.

Pri viackrokovej predikcii, začnime najprv 10-krokovou, sme znovu zhotovili tabuľku konečných hodnôt po desiatich epochách. Najlepšie modely, teda s najnižšími hodnotami, sme farebne rozlíšili a trénovali dlhšie. V tabuľke č. 6 sme označili znovu štyri najlepšie modely.

	mae	**Val_mae**			**mae**	**Val_mae**
256,2,2,2	0.4724	0.5132		**128,2,2,2**	0.4827	0.5307
256,2,2,4	0.4773	0.5263		**128,2,2,4**	0.4863	0.5327
256,2,4,2	0.4780	0.5116		**128,2,4,2**	0.4797	0.5195
256,2,4,4	0.4854	0.5189		**128,2,4,4**	0.4824	0.5174
256,4,2,2	0.4848	0.5295		**128,4,2,2**	0.4896	0.5335
256,4,2,4	0.4783	0.5375		**128,4,2,4**	0.4743	0.5100
256,4,4,2	0.4945	0.5391		**128,4,4,2**	0.4800	0.5203
256,4,4,4	0.4869	0.5256		**128,4,4,4**	0.4779	0.5261

Tabuľka č. 6: Hodnoty modelov Transformer po počiatočnom tréningu (10 epoch) – 10-kroková predikcia

20-krokovú predikciu sme spustili opäť na všetkých kombináciách vybraných parametrov a v tabuľke č. 7 sme zvýraznili modely s najnižšími hodnotami. Môžeme si všimnúť, že najlepšie hodnoty sú väčšie ako najlepšie výsledné hodnoty z 10-krokovej predikcie a jednokrokovej predikcie. Samozrejme, aj najlepšie výsledky z 10-krokovej predikcie sú väčšie ako najlepšie výsledky z jednokrokovej predikcie.

	mae	Val_mae			mae	Val_mae
256,2,2,2	0.5158	0.5455		**128,2,2,2**	0.5203	0.5539
256,2,2,4	0.5080	0.5328		**128,2,2,4**	0.5090	0.5389
256,2,4,2	0.5124	0.5542		**128,2,4,2**	0.5142	0.5480
256,2,4,4	0.5042	0.5427		**128,2,4,4**	0.5093	0.5382
256,4,2,2	0.5157	0.5438		**128,4,2,2**	0.5163	0.5434
256,4,2,4	0.5134	0.5517		**128,4,2,4**	0.5048	0.5505
256,4,4,2	0.5151	0.5441		**128,4,4,2**	0.5151	0.5509
256,4,4,4	0.5071	0.5391		**128,4,4,4**	0.5136	0.5469

Tabuľka č. 7: Hodnoty modelov Transformer po počiatočnom tréningu (10 epoch) – 20-kroková predikcia

V tabuľkách s najlepšími hodnotami sa nám niektoré modely opakujú. Samozrejme, s inými hodnotami, no stále medzi najlepšími natrénovanými. Iba jeden model vo všetkých troch prípadoch dosiahol veľmi dobrú hodnotu. Je to model s hodnotami head_size=128, num_heads=2, ff_dim=4, num_transformer_blocks=4 (v tabuľkách ako „128,2,4,4“).

Jeden model sa zhodol v dvoch prípadoch. V jednokrokovej a 10-krokovej predikcii dosiahol najlepšie výsledky, a to model s hodnotami head_size=256, num_heads=2, ff_dim=4, num_transformer_blocks=2 (v tabuľkách ako „256,2,4,2“).

Zvyšných sedem modelov sa vyskytlo iba jedenkrát, no aj tie sme spustili ešte raz a dlhšie. Konkrétne na päťdesiatich epochách hľadajúc najlepšie možné hodnoty v závislosti od dĺžky tréningu. Dôležitou vlastnosťou pri trénovaní modelov je aj sledovanie, v ktorej epoche hodnota val_loss, resp. val_mae začne byť väčšia ako hodnota loss, resp. mae. Všetky dlhšie trénované modely sme však nechali trénovať do konca päťdesiatej epochy. U všetkých modelov po tretej epoche, niekedy aj po druhej alebo štvrtej, bola hodnota val_loss väčšia ako bola hodnota loss.

V tabuľke č. 8 zobrazujeme výsledky z jednokrokovej predikcie po vyššie spomenutom dlhšom tréningu. Tabuľka č. 9 zobrazuje 10-krokovú predikciu a tabuľka č. 10 zobrazuje 20-krokovú predikciu. Farebne sú rozlíšené najnižšie hodnoty a príslušné modely.

Jednokroková predikcia – 50 epoch		
256,2,4,2	0.3313	0.4724
128,2,2,2	0.3329	0.4935
128,2,4,4	0.3367	0.4694
128,4,2,2	0.3444	0.4707

Tabuľka č. 8: Jednokroková predikcia (50 epoch)

10-kroková predikcia – 50 epoch		
256,2,2,2	0.3834	0.4697
256,2,4,2	0.3837	0.4710
128,2,4,4	0.3829	0.4738
128,4,2,4	0.3759	0.4688

Tabuľka č. 9: 10-kroková predikcia (50 epoch)

20-kroková predikcia – 50 epoch		
256,2,2,4	0.4036	0.4920
256,4,4,4	0.4042	0.4905
128,2,2,4	0.4021	0.4916
128,2,4,4	0.4035	0.4870

Tabuľka č. 10: 20-kroková predikcia (50 epoch)

Aby sme mohli vypočítať hodnoty MAE a MASE, museli sme výslednú predikciu naspäť denormalizovať. Hodnoty MAE a MASE sme vyhodnotili na len na jednokrokovej predikcii, pretože tie budeme porovnávať s článkom, kde sa venovali rovnakej jednokrokovej predikcii s inými modelmi. Z viackrokovej predikcie zobrazujeme hodnoty MAE len pre porovnanie v tabuľke č. 11.

10-kroková predikcia - MAE			20-kroková predikcia - MAE	
256,2,2,2	2823.0622		256,2,2,4	2968.8184
256,2,4,2	2811.7839		256,4,4,4	2967.8219
128,2,4,4	2890.6999		128,2,2,4	2929.0044
128,4,2,4	2848.5892		128,2,4,4	2948.9226

Tabuľka č. 11: Hodnoty MAE – viackroková predikcia

V tabuľke č. 12 zobrazujeme hodnoty MAE a MASE z jednokrokovej predikcie. Najlepší model budeme porovnávať s výsledkami z článku Network security situation awareness forecasting based on neural networks [57].

	MAE	MASE
256,2,4,2	2438.9031	1.4335
128,2,2,2	2455.1218	1.4331
128,2,4,4	2445.5441	1.4375
128,4,2,2	2417.0391	1.4207

Tabuľka č. 12: Hodnoty MAE, MASE – jednokroková predikcia

Podľa metrík MAE a MASE v tabuľke č. 12 vieme porovnať kvalitu vybraných modelov. Najnižšiu hodnotu MAE a zároveň aj MASE dosahuje model s hodnotami head_size=128, num_heads=4, ff_dim=2, num_transformer_blocks=2 (v tabuľkách ako „128,4,2,2“). Preto budeme ďalej s ostatnými modelmi porovnávať práve tento model.

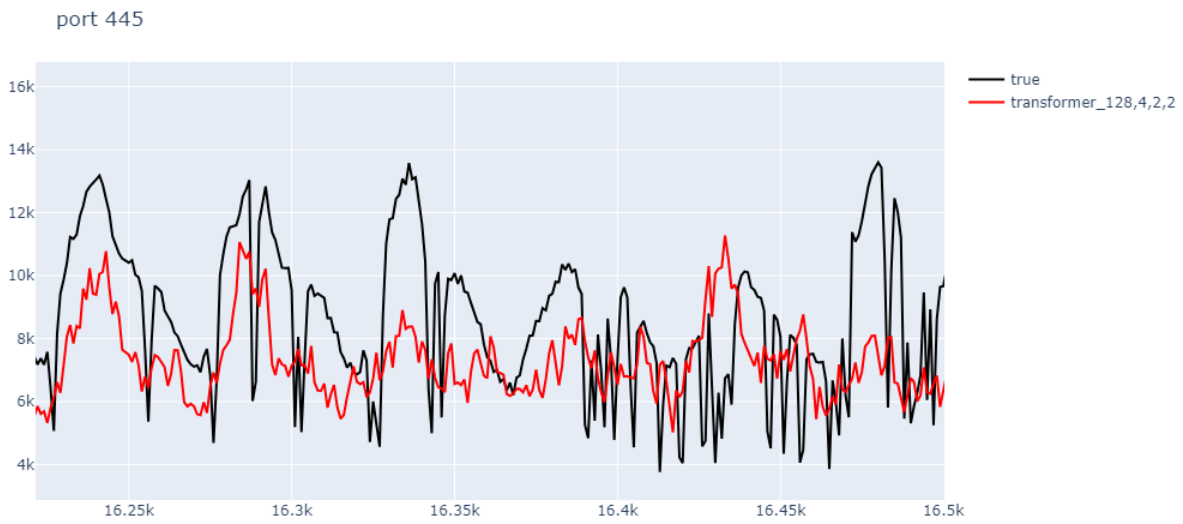
Porovnávať budeme s modelmi : Dense network (DN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), 1D convolution (Conv1D), Encoder-Decoder LSTM (e1d1) [57].

Test metrics	MASE	MAE
Loss function	**MAE**	**MAE**
DN	0.6972	1186.1808
LSTM	0.6633	1128.4426
GRU	0.6307	1072.9371
e1d1	0.6408	1090.1418
Conv1D	0.7080	1204.4519
Transformer	1.4207	2417.0391

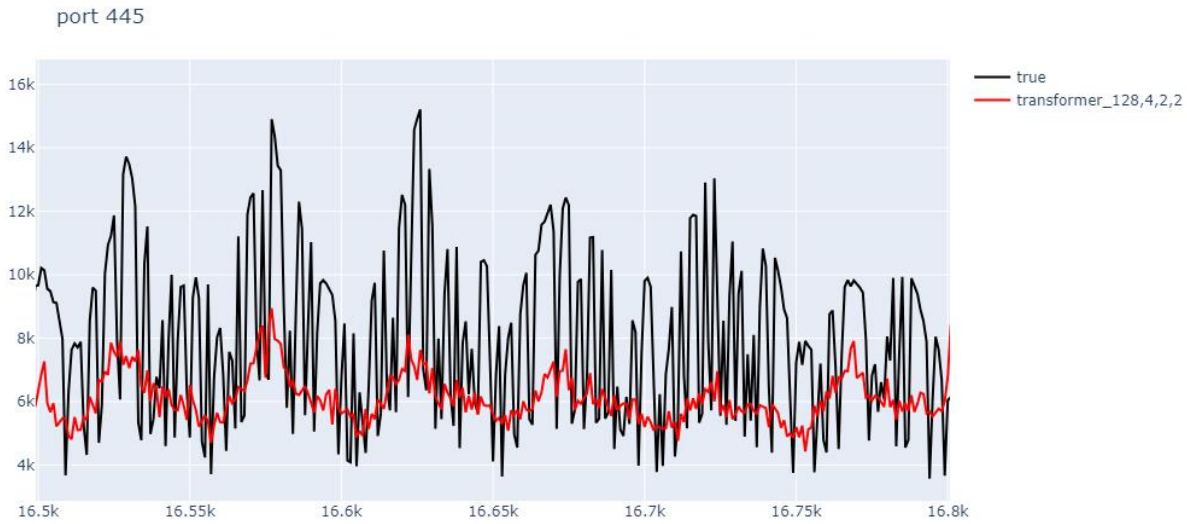
Tabuľka č. 13: Porovnanie hodnôt MAE a MASE modelu Transformer s hodnotami iných modelov prevzatých z článku Network security situation awareness forecasting based on neural networks [57].

V tabuľke č. 13 sme zobrazili hodnoty dosiahnuté v spomínanom článku a naše hodnoty s najlepším možným výsledkom. Farebne sú odlíšené najlepšie dosiahnuté výsledky. Model Transformer dosiahol najhoršie výsledky v porovnaní s ostatnými aj v metrike MAE a aj v MASE. Najlepšie výsledky dosiahol model GRU v metrike MAE aj MASE.

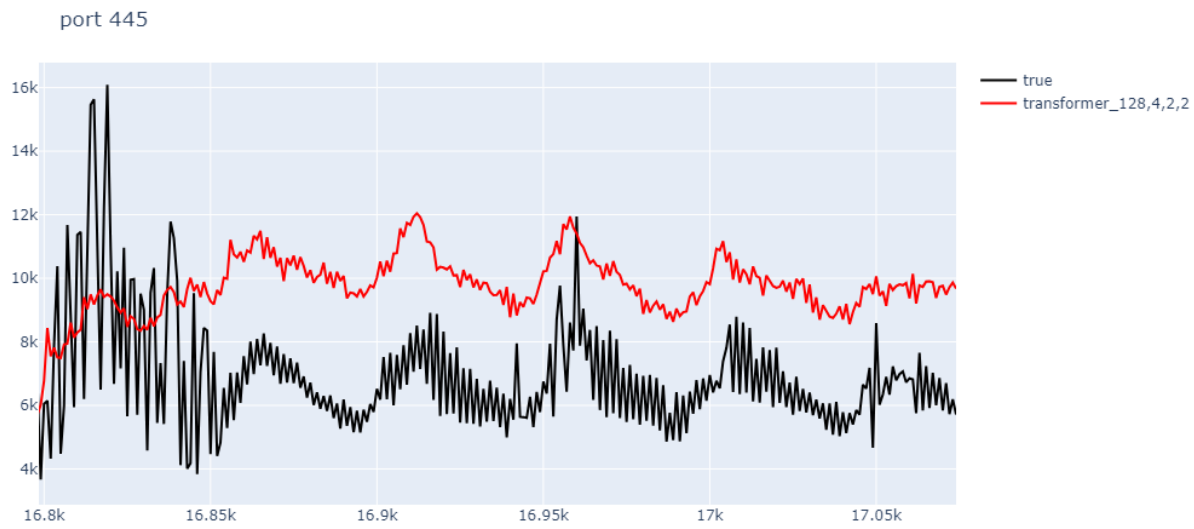
Na nasledujúcich obrázkoch č. 11, 12 a 13 znázorňujeme porovnanie predikcie Transformer. Čiernou farbou sú zobrazené skutočné hodnoty a červenou farbou predikované hodnoty. Naša predikcia nie je presnejšia ako spomínané ostatné modely. Už voľným okom vieme rozlíšiť, že naše predikované hodnoty sa celkom dosť líšia od skutočných hodnôt, a aj od predikovaných hodnôt iných modelov. Model Transformer je ešte celkom nový a určite bude potrebovať isté vylepšenia.



Obr. 11: Predikcia Transformer modelom “128,4,2,2” 1.časť

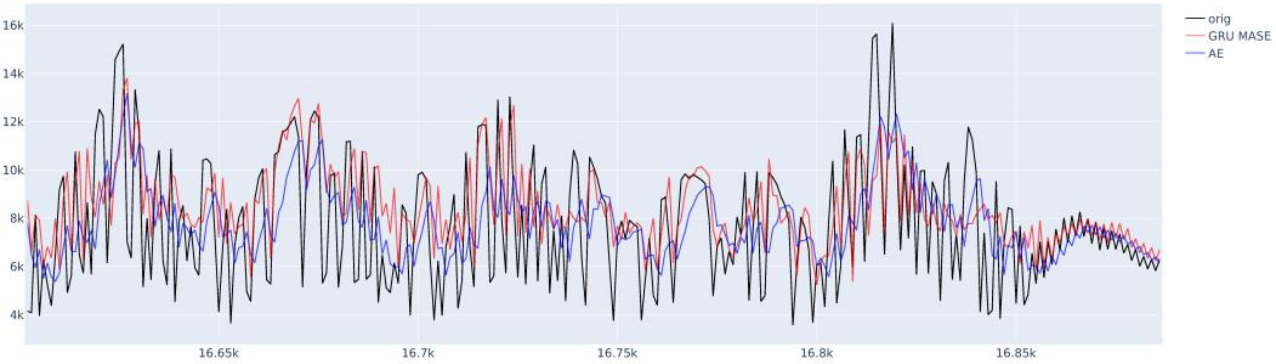


Obr. 12: Predikcia Transformer modelom “128,4,2,2” 2.časť

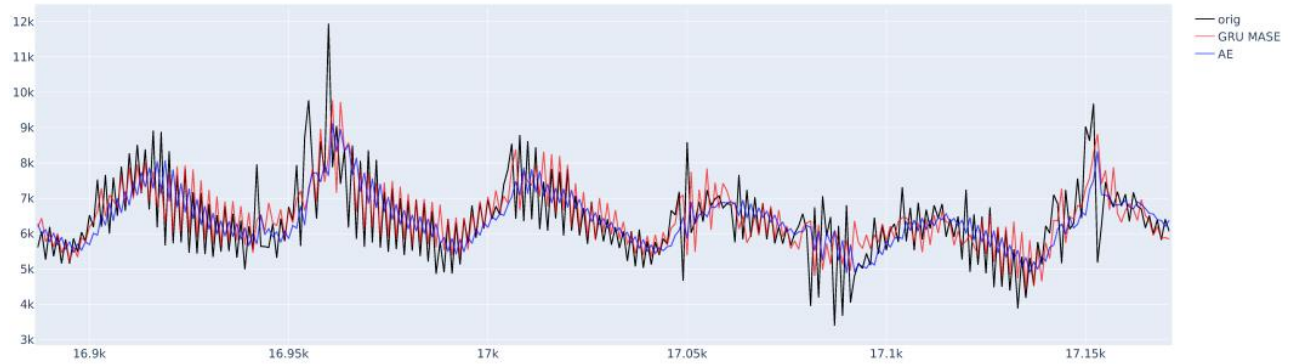


Obr. 13: Predikcia Transformer modelom "128,4,2,2" 3.časť

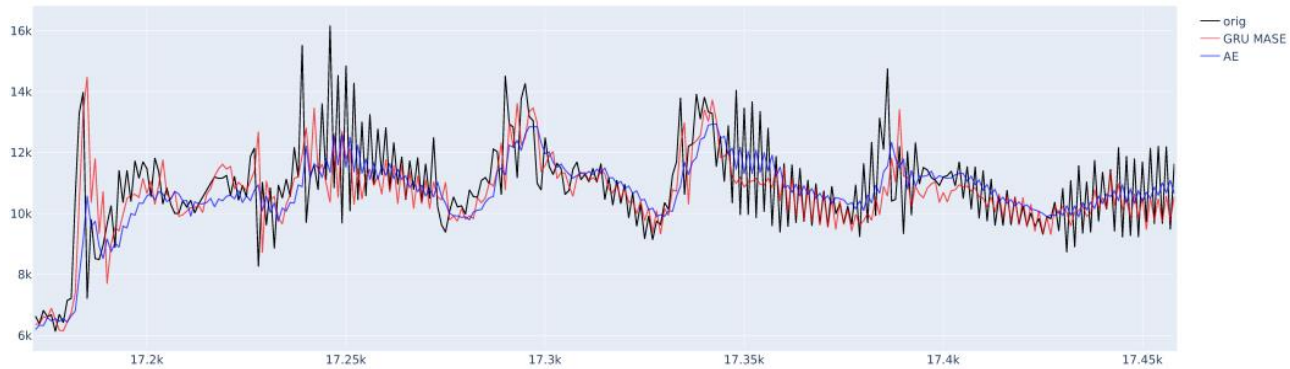
Pre porovnanie preberieme zo spomínaného článku graf s predikciami najlepších modelov. Na obrázkoch č. 14, 15 a 16 môžeme pozorovať, že ich hodnoty v každom bode sú veľmi blízke a v mnohých sa prelínajú. Čiernou farbou sú zobrazené skutočné hodnoty, červenou sú hodnoty z predikcie modelu GRU so sledovanou metrikou MASE a modrou farbou sú výstupy z predikcie štatistického modelu kombinovaného z modelov ARIMA a exponenciálnym vyhladzovaním.



Obr. 14: Predikcia GRU MASE a AE. 1. časť. Prevzaté z: [57]



Obr. 15: Predikcia GRU MASE a AE. 2. časť. Prevzaté z: [57]



Obr. 16: Predikcia GRU MASE a AE. 3.časť. Prevzaté z: [57]

Touto prácou sme však ukázali, že použitie modela Transformer je možné na predikciu časových radov v oblasti informačnej a kybernetickej bezpečnosti, no vzhľadom na to, že je to ešte celkom nový typ modelu, vyžaduje si ďalšie podrobné skúmanie a testovanie parametrov.

Záver

V práci sme sa zaoberali predikciou časových radov v oblasti informačnej bezpečnosti. Aktuálnosť tejto oblasti a existujúce pístupy predikcie sme rovinuli a analyzovali. Zamerali sme sa predovšetkým na neurónové siete, konkrétne model Transformer, ktorý je celkom nový medzi predikciami časových radov. Vypracovali sme jeden typ modelu Transformer s viacerymi parametrami a navzájom ich porovnali. Najlepší z nich sme porovnali aj s rôznymi modelmi z iných článkov.

Pozornosť tejto téme dnes venujú mnohé štúdie vzhľadom na pribúdajúce množstvo útokov a ich rýchlo meniace sa formy. Je potrebné stále dopĺňať túto oblasť novými poznatkami a vylepšovať už existujúce predikcie, resp. nachádzať nové prístupy predikcií, aby pomáhali včas upozorňovať na možné útoky, či už pre veľké obchodné spoločnosti, menšie spoločnosti, alebo aj bežných koncových používateľov internetu.

Prvým cieľom tejto práce bolo analyzovať existujúce prístupy predikcie časových radov v oblasti počítačovej bezpečnosti. To sme zrealizovali v úvodných dvoch kapitolách, ktoré pozostávali z definovania základných pojmov v oblasti predikcie, časových radov či rôznych prístupov predikcií. Uviedli sme aj príklady použitia jednotlivých prístupov, ich výhody a obmedzenia.

Po dokončení prvej časti sme sa zamerali na výber dát a implementovali sme model neurónovej siete na predikciu časových radov založený na modeli Transformer, čím sme naplnili druhý cieľ práce. Obsiahnutá je tu aj definícia základného modelu a opis naprogramovaného modelu. Vzájomné porovnávanie našich vybranných typov modelu Transformer je súčasťou poslednej, šiestej kapitoly.

V záverečnej podkapitole tejto práce sme sa venovali poslednému cieľu práce, t. j. porovnaniu dosiahnutých výsledkov s existujúcimi výsledkami. Opierali sme sa o článok, v ktorom sa venujú rovnakej predikcii časových radov, no s inými modelmi. Porovnanie ich výsledkov s našimi prinieslo všeobecné zhrnutie a poukázalo na najpresnejší model na vybraných dátach s rovnakým cieľom. Zároveň sa prejavila potreba vylepšovania modelu Transformer, nakoľko sa nám nepodarilo dosiahnuť dostatočné výsledky. Je potrebné dodatočné trénovanie s doplneným modelom, resp. s inými parametrami.

Zoznam použitej literatúry

1. Hyndman, R.J., Athanasopoulos, G. (2021). Forecasting: principles and practice, 3rd edition. Dostupné z: https://otexts.com/fpp3/
2. Ostertagová E. (2010). Modelling time series. *The13th International Scientific Conference Trends and Innovative Approaches in Business Processes 2010.* Dostupné z: https://www.sjf.tuke.sk/umpadi/taipvpp/2010/index.files/clanky%20PDF/OSTERTAGOVA.pdf
3. Vaswani A. a kolektív (2017). Attention Is All You Need. Dostupné z: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
4. Bahdanau D., Cho K., Bengio Y. (2015). Neural machine translation by jointly learning to align and translate. Dostupné z: https://arxiv.org/abs/1409.0473
5. I. Vallés-Pérez, E. Soria-Olivas, M. Martínez-Sober, A. J. Serrano-López, J. Gómez-Sanchís, F. Mateo (2022). Approaching sales forecasting using recurrent neural networks and transformers. Dostupné z: https://arxiv.org/abs/2204.07786
6. Bajtoš T., Gajdoš A., Kleinová L., Lučivjanská K., Sokol P. (2018). Network Intrusion Detection with Threat Agent Profiling. Dostupné z: https://www.hindawi.com/journals/scn/2018/3614093/
7. Zhao Z., Xia Ch., Chi L., Chang X., Li W., Yang T., Zomaya A. Y. (2021). Short-Term Load Forecasting Based on the Transformer Model. Dostupné z: https://www.mdpi.com/2078-2489/12/12/516
8. Negi S. (2021). Probabilistic Forecast of Time Series with Transformers and Normalizing Flows. Dostupné z: http://uu.diva-portal.org/smash/get/diva2:1627501/FULLTEXT01.pdf
9. M. Husák, M. Žádník, V. Bartoš, P. Sokol (2020). Dataset of intrusion detection alerts from a sharing platform. Dostupné z: https://www.researchgate.net/publication/347762333_Dataset_of_intrusion_detection_alerts_from_a_sharing_platform
10. Warden. [online] Dostupné z: https://warden.cesnet.cz/cs/index

11. What is an SMB Port + Port 445 and 139 explained. [online] Dostupné z: https://www.varonis.com/blog/smb-port

12. N. Wu, B. Green, X. Ben, S. O'Banion (2020). Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. Dostupné z: https://arxiv.org/abs/2001.08317

13. J. Brownlee (2018). Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.

14. Time series prediction with LSTM Recurrent Neural Networks in Python with Keras. [online] Dostupné z: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

15. How to develop LSTM Models for Time Series Forecasting. [online] Dostupné z: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

16. X. Fang, M. Xu, S. Xu, P. Zhao (2019). A deep learning framework for predicting cyber attacks rates. *EURASIP Journal on Information Security volume 2019, Article number: 5 (2019).* Dostupné z: https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-019-0090-6

17. M. Sykora (2020). Automatická klasifikace vybraných terénních tvarů z jejich kartografické reprezentace. *Diplomová práca.* Dostupné z: https://dspace.cuni.cz/bitstream/handle/20.500.11956/124473/120380278.pdf?sequence=1

18. M. Farkaš (2017). Predikcia spotreby energie pomocou hlbokých neurónových sietí. *Diplomová práca.* Dostupné z: https://www.itspy.cz/wp-content/uploads/2017/11/IT_SPY_2017_Diplomov_prce_12-1.pdf

19. How to develop Convolutional Neural Network Models for Time Series Forecasting. [online] Dostupné z: https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/

20. M. Lukáč (2016). Hlboké neurónové siete pre spracovanie multimédií. *Diplomová práca.* Dostupné z: https://is.muni.cz/th/r1pur/thesis.pdf

21. Convolutional Neural Networks (CNNs / ConvNets). [online] Dostupné z: https://cs231n.github.io/convolutional-networks/

22. Multi-Class Neural Networks: Softmax. [online] Dostupné z: https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax

23. A. Graves (2014). Generating Sequences With Recurrent Neural Networks. *Department of Computer Science University of Toronto.* Dostupné z: https://arxiv.org/pdf/1308.0850.pdf

24. Y. -B. Leau, S. Manickam (2015). Network Security Situation Prediction: A Review and Discussion. *ICSIIT 2015: Intelligence in the Era of Big Data, pp 424–435.* Dostupné z: https://link.springer.com/chapter/10.1007/978-3-662-46742-8_39

25. H. Bilgil (2021). New grey forecasting model with its application and computer code. *AIMS Mathematics 2021, Volume 6, Issue2: 1497-1514.* Dostupné z: https://www.aimspress.com/article/id/5fb7afd4ba35de5c8625316d

26. J. K. Blitzstein, J. Hwang (2015). Introduction to Probability. Dostupné z: https://www.statisticiansforhire.com/wp-content/uploads/2019/04/introduction-to-probability-by-joseph-k-blitzstein-and-jessica-hwang.pdf

27. H. Changlin, L. Yufen (2017). Survey of Network Security Situation Awareness. *2017 International Conference on Computational Science and Engineering (ICCSE 2017).* Dostupné z: https://www.atlantis-press.com/article/25884992.pdf

28. Neural machine translation with a Transformer and Keras. [online] Dostupné z: https://www.tensorflow.org/text/tutorials/transformer

29. How to code The Transformer in Pytorch. [online] Dostupné z: https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ec

30. Text classification with Transformer. [online] Dostupné z: https://keras.io/examples/nlp/text_classification_with_transformer/

31. Training and validation Loss in Dep Learning. [online] Dostupné z: https://www.baeldung.com/cs/training-validation-loss-deeplearning

32. Mean Absolute Error – MAE [Machine Learning (ML)]. [online] Dostupné z: https://medium.com/@20__80__/mean-absolute-error-mae-machine-learning-ml-b9b4afc63077

33. Dostupné z: https://www.researchgate.net/profile/Nura-Aljaafari/publication/332092821/figure/fig4/AS:779719519764482@1562911028330/Example-of-max-pooling-and-average-pooling-operations-In-this-example-a-4x4-image-is.jpg

34. N. R. Pokhrel, Ch. P. Tsokos (2017). Cybersecurity: A Stochastic Predictive Model to Determine Overall Network Security Risk Using Markovian Process. *Journal of Information Security, Vol.8 No.2, April 2017.*

35. L. Nian, L. Geng, L. Yong (2011). A Method Of Network Security Situation Prediction Based on Gray Neural Network Model. *Applied Mechanics and Materials, vol. 63, pp. 936–939.*

36. Time series forecasting. [online] Dostupné z: https://www.tensorflow.org/tutorials/structured_data/time_series

37. Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37-45.

38. J. C. Chambers, S. K. Mullick, D. D. Smith (1971). How to Choose the Right Forecasting Technique. [online] Dostupné z: https://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique

39. A. M. Davey, B. E. Flores (1993). Identification of seasonality in time series: A note. *Mathematical and Computer Modelling, vol. 18, issue 6, pp. 73-81*

40. Prediction. [online] Dostupné z: https://h2o.ai/wiki/prediction/

41. A. Haluszczynski, Ch. Räth (2019). Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing. Dostupné z: https://arxiv.org/abs/1907.05639

42. J. Fattah, L. Ezzine, Z. Aman, H. El Moussami, A. Lachhab (2018). Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management, vol. 10.*

43. A. Shameli-Sendi, M. Dagenais, M. Jabbarifar, M. Couture (2012). Real Time Intrusion Prediction based on Optimized Alerts with Hidden Markov Model. *Journal of Networks 7(2).*

44. H. A. Kholidy, A. Erradi, S. Abdelwahed (2015). Attack Prediction Models for Cloud Intrusion Detection Systems. *Conference: Proceedings – 2nd International Conference on Artificial Intelligenc, Modelling, and Simulation, AIMS 2014.*

45. H. A. Kholidy, A. Erradi, S. Abdelwahed, A. Azab (2014). A Finite State Hidden Markov Model for Predicting Multistage Attacks in Cloud Systems. *Conference: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (DASC)*

46. H. A. Kholidy, A. Yousof, A. Erradi, S. Abdelwahed (2015). A Finite Context Intrusion Prediction Model for Cloud Systems with a Probabilistic Suffix Tree. *Conference: 2014 European Modelling Symposium (EMS).*

47. Y. B. Leau, S. Manickam (2016). A Novel Adaptive Grey Verhulst Model for Network Security Situation Prediction. *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 1, 2016.*

48. Y. B. Leau, S. Manickam (2016). An Enhanced Adaptive Grey Verhulst Prediction Model for Network Security Situation. *IJCSNS International Journal of Computer Science and Network Security, Vol. 16, No. 5, 2016.*

49. Zheng, R., Zhang, D., Wu, Q., Zhang, M., Yang, C. (2012). A Strategy of Network Security Situation Autonomic Awareness. *Network Computing and Information Security. NCIS 2012. Communications in Computer and Information Science, vol 345.*

50. Fenglan Ch., Yongjun S., Guidong Z., Xin L. (2013). The network security situation predicting technology based on the small-world echo state network. *IEEE 4th International Conference on Software Engineering and Service Science*, Beijing, China, 2013, pp. 377-380

51. Zhang, Y., Jin, S., Cui, X., Yin, X., Pang, Y. (2013). Network Security Situation Prediction Based on BP and RBF Neural Network. *ISCTCS 2012. Communications in Computer and Information Science, vol 320. Springer,* Berlin, Heidelberg.

52. W. Xing-zhu (2016). Network Intrusion Prediction Model based on RBF Features Classification. *International Journal of Security and its Applications 10 (4), pp. 241-248.*

53. M. Husák, J. Komárková, E. Bou-Harb, P. Čeleda (2018). Survey of Attack Projection, Prediction, and Forecasting in Cyber Security. Dostupné z: https://is.muni.cz/publication/1434138/2019-COMST-survey-of-attack-projection-prediction-forecasting.pdf

54. M. H. Sazli (2006). A brief review of feed-forward neural networks. *Communications Faculty Of Science University of Ankara, 50 (1), pp. 11-17.*

55. LSTM Vs GRU in Recurrent Neural Network: A Comparative Study. [online] Dostupné z: https://analyticsindiamag.com/lstm-vs-gru-in-recurrent-neural-network-a-comparative-study/

56. Dostupné z: https://en.wikipedia.org/wiki/Gated_recurrent_unit#/media/File:Gated_Recurrent_Unit,_base_type.svg

57. R. Staňa, P. Pekarčík, A. Gajdoš, and P. Sokol (2021). Network security situation awareness forecasting based on neural networks

58. P. J. Brockwell, R. A. Davis (1991). Time Series: Theory and Methods, Second Edition. ISBN 978-1-4419-0319-8

59. M. A. Erlinger, M. Wood (2007). Intrusion Dtection Message Exchange Requirements. *RFC – Informational.*

60. The Basics of Neural Networks (Neural Network Series) – Part 1. [online] Dostupné z: https://towardsdatascience.com/the-basics-of-neural-networks-neural-network-series-part-1-4419e343b2b

Prílohy

Príloha A: Všetky zdrojové kódy, ktoré boli použité pri tvorbe tejto práce sú dostupné na: https://gitlab.science.upjs.sk/Barbora.Fedova/bakalarska_praca