VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

Ing. Václav Oujezský

KONVERGOVANÉ SÍTĚ A TOMOGRAFIE SÍŤOVÉHO PROVOZU S VYUŽITÍM EVOLUČNÍCH ALGORITMŮ

CONVERGED NETWORKS AND TRAFFIC TOMOGRAPHY BY USING EVOLUTIONARY ALGORITHMS

ZKRÁCENÁ VERZE PH.D. THESIS

Obor: Teleinformatika
Školitel: doc. Ing. Vladislav Škorpil, CSc.
Oponenti:

Datum obhajoby:

Klíčová Slova

Behaviorální analýza sítě, evoluční algoritmus, Python, síťová sonda, tomografie síťového provozu.

Key Words

Network Behavior Analysis, Evolutionary Algorithms, Network Probe, Python, Network Tomography.

Dizertační práce je k dispozici na Vědeckém oddělení děkanátu FEKT VUT v Brně, Technická 10, Brno, 616 00

ISBN 80-214-
ISSN 1213-4198

Obsah

1 Úvod a analýza současného stavu

Dizertační práce se věnuje tématu z oblasti konvergovaných sítí, zaměřuje se na problematiku síťové tomografie, respektive monitoringu a zpracování komunikačních dat s využitím evolučních algoritmů. Vlastní výzkum volně navazuje na projekty „Detekce bezpečnostních hrozeb na aktivních prvcích kritických infrastruktur“ a „Redukce bezpečnostních hrozeb v optických sítích“.

1.1 Úvod

Současné konvergované sítě umožňují vzájemnou kooperaci na výkonové, výpočetní a funkční úrovni. Moderní koncept sítí umožňuje nasazení sofistikovaných metod k řízení provozu, jejich konfiguraci a zabezpečení. Otázka bezpečnosti konvergovaných sítí nabývá stále na významu. Nevyžádaný provoz výrazně zasahuje do poskytovaných služeb zákazníkům. V zásadě již nezáleží na použitých nástrojích, ale na způsobu provedení síťových útoků.

Současným trendem je použití hybridního přístupu, kdy útočníci typicky mixují několik operací dohromady k vytvoření několika vektorů útoku. Během takového jednání je měněn vektor útoku a signatury protokolů za účelem oklamání automatizovaných mitigačních zařízení, která jsou navíc používána decentralizovaně a různorodě. Nejefektivnější síťové útoky využívají předem získaných znalostí z otisků (*footprint*) jednotlivých automatizovaných mitigačních zařízení, která se poté stávají neúčinnými. Vývoj v konvergenci sítí přináší nová bezpečnostní rizika, jako jsou například útoky „Denial of Energy“ nebo „Denial of Company Reachability“.

Síťová tomografie je disciplína, která studuje interní chování a charakteristiku datového provozu a sítě pomocí externích zařízení, koncových bodů. Tato zařízení mohou být reprezentována specializovanými hardwarovými sondami a jednotlivými prvky, jako jsou směrovače, počítače, mobilní zařízení a technologie IoT (*Internet of Things*). Všechna tato zařízení mohou poskytovat data pro účely analýzy.

V informatice se anomálie označuje jako odlehlá hodnota a data, která se výrazně liší od ostatních, či referenčních dat se nazývají v angličtině obecně „outlier“. Anomálie síťového provozu mohou být rozlišovány v závislosti na jejich výskytu a původu. Jak uvádí autor Eduardo B. Fernandez [1], základními technikami používanými v praxi jsou použití algoritmů výpočetní inteligence, verifikace protokolů či statistického modelování.

Systémy používající modifikované genetické algoritmy vyhovují současným požadavkům v oblasti rozhodovacího procesu a přistupují k modelování chování síťových protokolů a aplikací tak, že zavádí rekurzi odlišných stavů provozu, jako je „normální“ chování, chybový stav a stav útoku.

Pro výše popsané techniky je důležitý použitý zdroj dat pro samotnou analýzu. Jako zdroj je možné použít informace o síti a klientech získané skenováním sítě. Skenování sítí je možné rozdělit do dvou hlavních kategorií, a to pasivní sledo-

vání sítě a aktivní skenovaní sítě. Do skupiny pasivního sledování patří zařízení, která jsou vsazena do sítě a pasivně naslouchají na otevřených portech, či poskytují službu, která je určena pro účely detekce.

Jelikož je dané téma vlastní práce velice obsahově široké, bylo nutné se zaměřit na specifickou oblast daného výzkumu. Konvergované sítě obecně představují rychle se rozvíjející oblast. Pro vlastní účely práce byl zvolen princip sběru dat pomocí protokolu NetFlow [2], a to také s ohledem na možnosti a vybavení laboratoře, kde byl výzkum prováděn.

Problematika bezpečnosti sítí stále narůstá na významu a konvergované sítě jsou nedílnou součástí této tématiky. Do této oblasti jsou zařazeny problematiky predikce, detekce a obrany. Z pohledu bezpečnosti sítí byla zvolena problematika detekce, která je základním předpokladem pro řešení predikce a obrany. Z oblasti bezpečnostní detekce byla zvolena tématická sekce detekce anomálií. Ač anomálie mohou představovat jakékoliv tendence, odchylky či směry, práce se soustředí na možnosti detekce anomálií jako takových a možnosti nasazení genetických algoritmů. Z pohledu neprobádaných směrů je práce soustředěna na životnost a životní cyklus zvoleného anomálního jevu.

1.2 Analýza současného stavu řešené problematiky

První výrazné zmínky o síťové tomografii (*Network Tomography*) byly uvedeny autorem Vivardi [3]. V tomto díle bylo snahou zachytit vztah mezi maticí výchozího provozu (*Origin Destination Matrix*) a počtem propojení jednotlivých uzlů sítě.

Autoři Przemysław Berezinski a kolektiv [4] porovnávali jednotlivé druhy přístupu detekcí malware pomocí entropie ve vzorcích dat sítě. Porovnávali schopnosti detekce při použití entropie podle Shannona, Rényi a Tsallise. Došli k závěru, že detekce moderních botnetových sítí [3] na základě entropie je proveditelná. Nejlepší výsledky podávaly výpočty dle Rényi a Tsallise.

Analýza hlavních komponent PCA (*Principal Components Analysis*) [5] představuje cestu, jak identifikovat vzory v sadě dat s vysokou dimenzí a zvýraznit jejich odlišnosti a podobnosti. Slouží také ke snížení dimenze samotných dat. Jelikož jsou nalezeny tyto vzory, je možné provést kompresi. PCA představuje transformaci do souřadnic, které mapují sadu n dimenzionálních datových bodů na n nových, nekorelovaných proměnných, které se nazývají hlavní komponenty. Provozní zátěž síťových linek ma nízkou efektivitu dimenzí [6], pomocí PCA analýzy lze tudíž efektivně detekovat anomálie.

V oblasti počítačové bezpečnosti jsou také používány algoritmy vycházející z biologicky inspirovaných metod. Patří sem zejména neuronové sítě, výpočty pomocí evolučních algoritmů a umělé imunitní systémy AIS (*Artificial Immune Systems*). Pro detekci anomálií jej použil Dasgupta (1996), pro rozpoznávání vzorů dat autoři

[3] *Botnet* – je kombinace slov robot a síť. Jedná se o velmi obtížně identifikovatelný provoz, většinou určený pro šíření škodlivého kódu a provedení DDoS (*Distributed Denial of Service*) útoku.

Forest (1993), Gibert (1994). K těžení dat byla tato metoda použita autory Huntem (1996) a Timmisem (2001–2002). Třemi základními popsanými technikami (mechanizmy) jsou: teorie imunitní sítě, záporný selekční mechanizmus a princip klonální selekce. Dalšími jsou Bone-marrow model, afinitní[4] funkce a somatická hypermutace.

Genetické algoritmy byly použity pro vylepšení stávájících metod detekce, či predikce anomálií. Autoři Divya Somvanshi a R.D.S. Yadava [7] použili GA v PCA analýze pro extrakci komponent. Autor Wilson Rivera-Gallego [8] využívá genetického algoritmu pro výpočet matic Euklidovské vzdálenosti.

Síťová anomografie (*Network Anomography*) – autorů Yin Zhang a kolektiv [9]. Byl zde navržen algoritmus pro prostorovou detekci anomálií pomocí síťové tomografie, kterou nazvali „anomografie" spojením slov tomografie a anomálie. Nabízí ucelenou myšlenku využití síťové tomografie k detekci a odvození anomálií. Stejně jako PCA používá prostorová detekční schémata a jako statistická analýza používá dočasná schémata.

Autoři definují problém odvození anomálií z nepřímých měření linek SD (*Source to Destination*), protože anomálie nemohou být v mnoha případech měřeny přímými metodami. Navrhli algoritmus, který sleduje směrování a provoz v síti. Tento algoritmus je schopný zacházet se změnami ve směrování a chybějícími daty v měření. Pro samotnou evaluaci výsledků použili síť Abilene a síť ISP Tier-1.

Metoda ASTUTE (*A Short-Timescale Uncorrelated-Traffic Equilibrium*) byla prezentována autory Fernando Silveira, Christophe Diot, Nina Taft a Ramesh Govindan [10] jako metoda schopná detekovat různé typy síťových anomálií v mnoha malých datových tocích, oproti použití algoritmu Kalmanova filtru v málo velkých datových tocích, který provádí bodový odhad stavů na základě zašuměných výstupů z měření.

Autoři projektu IPS Stratosphere [11] ČVUT v Praze, Sebastian Garcìa a kolektiv, využívají Markovovské řetězce pro behaviorální detekci anomálií.

Pro výše popsané techniky je důležitý použitý zdroj dat pro samotnou analýzu. Jako zdroj je možné použít informace o síti a klientech získané skenováním sítě. Jednou z možností je využití protokolu NetFlow. Byl vyvinut společností Cisco Systems, Inc. a implementován v jejich síťových produktech [2]. Tento protokol je velmi populární a v různých variantách je používán také mnoha jinými výrobci síťových prvků. Nejnovějším nástupcem je Internet Protocol Flow Information Export (IPFIX). Zpráva protokolu IPFIX se skládá z hlavičky a pole obsahující záznamy o provozu. Tyto zprávy jsou zasílány ze síťových zařízení do analyzátorů (kolektorů). Příkladem použití NetFlow je uchovávání dat „provozních údajů" s názvem Data Recognition (DR), což je vyžadováno nejen českou legislativou, ale také například evropskou legislativou. V České republice (ČR) se jedná o „zákon o elektronických komunikacích" (č. 127/2005 Sb), konkrétně § 97 odstavec 3. Tento zákon byl založen původně na směrnici Rady 2006 Evropského parlamentu a Rady/24/ES.

[4]Afinita – síla vazby ligandu (molekuly) ke svému receptoru.

Tato směrnice byla zrušena v roce 2013. Evropská legislativa například definuje DR v doporučení R (87) 15, užití osobních dat v policejním sektoru.

Zařízení určená pro sběr výše uvedených dat se nazývají obecně kolektory. Příkladem kolektoru je Scrutinizer [12], či PRTG systém [13]. Jsou–li síťová zařízení správně nakonfigurována, jsou pravidelně odesílány informace o tom, které uzly (IP adresy) komunikují. Tyto zprávy obsahují také informace o portech, protokolech a také informace o délce trvání jednotlivých spojení

Dle zkoumaných faktů je pro analýzu anomálií dat v reálném čase výhodné použít kombinaci známých postupů a systémů k tomu určených. Zařízení k detekci anomálií síťového provozu jsou označována souhrnným názvem IDS, (*Intrusion Detection System*). Tyto systémy mohou být rozděleny do jednotlivých skupin podle používaných technik. A to do skupin abstraktních, signaturních a chování.

Důležitým aspektem a mezikrokem samotné analýzy je těžení dat a klastrování dat. Mnoho multivariantních technik (vícerozměrných analýz) a modelů aplikovaných na detekci anomálií jsou založeny na konceptu vzdálenosti. Nejznámější metrikou je Euklidovská vzdálenost. Jako taková je hojně používána pro měření spojitostí či similarity (podobnosti). Jsou–li brány v úvahu dva rozdílné vektory $\mathbf{x} = (x_1, x_2, \dots, x_n)$ a $\mathbf{y} = (y_1, y_2, \dots, y_n)$ jako dvou dimenzionální zjištění hodnot měření, potom Euklidovská vzdálenost mezi $\mathbf{x}$ a $\mathbf{y}$ je definována dle vztahu (1):

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} \tag{1}$$

Protože každá hodnota vlastního vektoru přispívá co do výpočtu Euklidovské vzdálenosti, mohou se výsledky výrazně lišit i při malé změně hodnot. Hraje zde roli i dominance jedné sady hodnot vůči druhé. Proto se variabilita dá zanést přímo do výpočtu. Jednou z nejznámějších těchto metrik je Mahalanobisova vzdálenost (2), kde $\mathbf{V}$ představuje váženou disperzní matici – kovariancí parametrů.

$$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{V}^{-1} (\mathbf{x} - \mathbf{y}) \tag{2}$$

Matice kovariancí parametrů je zobecnění pojmu rozptylu pro náhodné vektory. Pokud je tato matice identickou maticí, je Mahalanobisova vzdálenost redukována na Euklidovskou vzdálenost. Pokud je tato matice diagonální, pak se jedná o normalizovanou Euklidovskou vzdálenost.

Ke třídění získaných dat z vícerozměrných pozorování jsou využívány techniky shlukování. Data jsou setříděna tak, aby se rozdíl hodnot dat členů skupiny blížil k nule. Shluková analýza se zabývá právě tvorbou takovýchto homogenních celků. Je snížen počet dimenzí dat a jedna proměnná vyjadřuje příslušnost datové jednotky ve shluku.

Postup shlukování je možné popsat obecně tak, že máme k dispozic datovou matici $\mathbf{X}_{(m,n)}$, kde m je počet objektů a n je počet proměnných. Počet shluků je značen k. Jedná se o rozklad množiny m objektů v závislosti na hodnotách n do k shluků. V potaz jsou brány pouze rozklady s disjunktními shluky. Jeden objekt

musí patřit pouze jednomu shluku S_k. Je vypočtena vzdálenost pro všechny objekty. Z tohoto výpočtu vznikne symetrická čtvercová matice zvaná asociační matice.

Metody shlukování jsou rozdělovány na hierarchické nebo nehierarchické, podle struktury dat. Nehierarchické shlukování je vhodné pro velké objemy dat, mezi něž patří metoda K-průměrů, metoda X-průměrů a metoda K-medoidů. Následně je uveden princip metody K-průměrů. Tato metoda se považuje mezi výše uvedenými za jejich základ.

1. Data jsou náhodně rozdělena do k shluků.
2. Je určeno k centroidů[5] c_k pomocí konceptu průměru vzdálenosti ve shluku.
3. Je hodnocen každý objekt shluku a jeho vzdálenost k centroidu. Pokud má blíže k jinému, je přemístěn a centroidy jsou opět přepočítány tak, že je vypočítán nový průměr ze všech prvků shluku.
4. Je opakován předchozí bod do doby, kdy žádný z prvků již není možné přemístit.

Matematicky je možné vyjádřit vztah k shluků S_k a k centroidů c_k minimalizací S_k a c_k dle následujícího vztahu (3).

$$\sum_{k=1}^{k} \sum_{x_n \in S_k} \|x_n - c_k\|^2 \tag{3}$$

Problém minimalizace představuje těžkou úlohu řešení. Nejznámějším řešením je pomocí Lloydova algoritmu. Jakmile jsou známy centroidy, jsou prvky přiřazeny dle koncepce vzdálenosti dle následujícího vztahu.

$$S_k = \{x_n : \|x_n - c_k\| \leq \forall \|x_n - c_k\|\},$$
$$c_k = \frac{1}{S_k} \sum_{x_n \in S_k} x_n \tag{4}$$

Existuje několik modifikací tohoto algoritmu. Mezi nevýhody patří pevná definice k shluků a využití výpočtu pomocí Euklidovské vzdálenosti, který je náchylný na vzdálené objekty. Pro validaci počtu k shluků existují opět metody jejich validace, jako je validační metoda siluety či Daviesův-Bouldinův validační index DB (*Davies-Bouldin Validity Index*). Daviesův-Bouldinův validační index vychází z podílu sumy rozložení uvnitř shluku a rozložení mezi shluky. Tento index je získán ze vzorce (5).

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left\{ \frac{S_n(Q_i) + S_n(Q_j)}{S_n(Q_i, Q_j)} \right\}, \tag{5}$$

[5]Centroid je střed shluku. Jedná se o vektor obsahující průměry proměnných pozorovaných ve shluku.

kde n je počet shluků, $S_n(Q_i)$ je průměrná vzdálenost uvnitř shluku od jeho středu a $S_n(Q_i, Q_j)$ je vzdálenost mezi jednotlivými shluky reprezentované centroidy.

1.3 Evoluční algoritmy

Uvedené poznatky pochází zejména z literatury autorů McDonnell a kol. [14] a Hynek, J. [15]. Problematika evolučních algoritmů je velice obsáhlá a evoluční algoritmy jsou stále předmětem výzkumu. Počet publikací v této oblasti stále roste. Evoluční algoritmy vycházejí zejména z principů evoluční teorie o původu druhů přírodním výběrem, či uchováním prospěšných plemen v boji o život, „On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life“ Charlese Roberta Darwina (rok 1859). Dále také zakladatele moderní genetiky Johanna Gregora Mendela, rodáka z Dolního Slezska, z jeho práce o experimentech na rostlinách „Versuche über Pflanzenhybriden“ (rok 1856). Prvními pionýry v této oblasti byli Fraser, Bremermann a Reed v 50tých a 60tých létech dvacátého století.

Evoluční algoritmy jsou zařazeny do oblasti řešení metaheuristik s populacemi (*Population-based Metaheuristics*). Tento problém vyžaduje najít řešení nestatických proměnných. Mají za cíl najít takové řešení $\bar{X}$, že provádí optimalizaci funkce, jak uvádí výraz (6).

$$\begin{gathered}\text{optimalizace } \bar{X}, \text{kde } \bar{X} = (x_1, x_2, \dots, x_n) \in \mathcal{R}^n \\ \text{a } \bar{X} \in \mathcal{F} \subseteq \mathcal{S}\,,\end{gathered} \tag{6}$$

přičemž množina $\mathcal{S} \in R^n$ definuje prohledávaný prostor a množina $\mathcal{F} \subseteq \mathcal{S}$ definuje nejvíce vyhovující prostor z prostoru prohledávaného. Většinou je prohledávaný prostor $\mathcal{S}$ definován jako n dimenzionální prostor v $\mathcal{R}^n$, proměnné jsou definovány jako dolní a horní hranice dle výrazu (7), $L =$ levá strana, $P =$ pravá strana.

$$L(i) \leq x_i \leq P(i);\ 1 \leq i \leq n \tag{7}$$

Množina $\mathcal{F}$ je definována na prohledávaném prostoru množiny $\mathcal{S}$ a jsou přidány dodatečné omezující podmínky uvedené v (8).

$$\begin{gathered}g_j(\bar{X}) \leq 0; pro\ j = 1, \dots, q \\ h_j(\bar{X}) = 0; pro\ j = q+1, \dots, m\end{gathered} \tag{8}$$

Obecně, evoluční techniky používají k ohodnocení (evaluaci) nejlepšího řešení (jedince) účelovou (*objectives*) funkci f, též nazývanou fitness funkce, (9).

$$eval_f(\bar{X}) = f(\bar{X}); pro \bar{X} \in \mathcal{F} \tag{9}$$

Dále jsou používány omezující podmínky f_j pro j-tou podmínku pro konstrukci ohodnocení. Tato funkce je definována vztahem (10).

$$f_j(\bar{X}) = \begin{cases} \max\{0, g_j(\bar{X}) & \text{pokud } 1 \leq j \leq q \\ |h_j(\bar{X})| & \text{pokud } q+1 \leq j \leq m \end{cases} \quad (10)$$

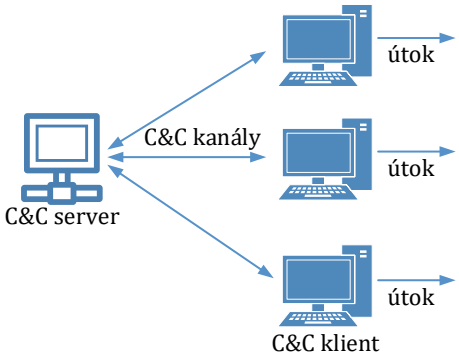
Nejpoužívanější evoluční algoritmy jsou algoritmy genetické a kombinované evoluční strategie (ES). Evoluční algoritmy se používají jak pro jednokriteriální optimalizaci, tak také pro vícekriteriální optimalizaci, nazývanou také multi-objektivní či více-objektivní optimalizace MOO (*Multi-Objective Optimization*). Inspirovány biologií, evoluční algoritmy převzaly pojmy jedinec, populace a fitness funkce. Jedinec představuje přípustné řešení, skupina jedinců populaci. Ohodnocující funkce určuje kvalitu jedinců a jedná se o optimalizační funkci, kde je hledáno globální maximum nebo minimum.

1.4 Popis problému

Podnětem pro práci samotnou jsou existující problémy spojené s behaviorální analýzou dat v konvergovaných sítích. Specifický výzkum je zaměřen na vývoj algoritmu pro detekci anomálií. V současnosti tvoří provoz botnetových sítí bezpečnostní slabinu. Z těchto sítí jsou prováděny útoky či sdílení nebezpečných obsahů. Tento provoz je nutné vnímat jako anomálii, kterou je nezbytné analyzovat a specifikovat.

Jednotlivé botnet sítě jsou rozlišovány dle jejich vzhledu a typu použitého protokolu pro komunikaci. Autor Silvia a kolektiv [17] definoval jednotlivé typy botnet sítí a jejich chování. Jeden z nejstarších typů této sítě využívá pro komunikaci protokol IRC (*Internet Relay Chat*). Mezi další typy se řadí: HTTP (*Hypertext Transfer Protocol*), P2P (*Point to Point*) a HTTP2P provoz v kombinaci s centralizovanou a decentralizovanou správou.

Výčet výše uvedených kategorií není jediný. Obecně platí, že botnetové sítě mohou být řízeny specifickým řídícím serverem, který využívá vlastní typ přístupu, neboli získání „root“ práv síťového zařízení. Tento přístup může být zajištěn například pomocí SSH (*Secure Shell*) připojení.



Obr. 1: Jednoduchá centralizovaná síť botnet

Vlasntní C&C (*Command and Control*) kanály zajišťují příjem a odesílání řídicích příkazů a informací mezi řídicím C&C serverem a infikovanými klienty, jak je znázorněno na obrázku 1. Řídicí server je schopen řídit mnoho klientů v krátkém časovém období. Na základě příkazů řídicího serveru je možné provést masivní útok typu DDoS (*Distributed Denial of Service*).

Vzhledem k výše uvedenému, je rozlišováno mnoho druhů stávajících přístupů a metod, jak takový botnet vytvořit. Je nezbytné říci, že proces převzetí kontroly nad síťovými zařízeními není pevně definován a může být proveden v zásadě jakýmkoliv individuálním přístupem, a to i pomocí naprogramování vlastního řešení. Takovéto řešení je poté pro ostatní neznámé a nepředvídatelné.

Detekce botnetu je obecně založena na metodikách chování nebo popisu C&C infrastruktury. Problém této detekce je v tom, že provoz samotného botnetu se mísí s ostatním provozem a chování tohoto škodlivého provozu může být podobné jako chování provozu „normálního“.

Pokud je pro detekci použit systém IDS (*Intrusion Detection System*), je zde opět problém se šifrováním mezi C&C uzly a vzory či modely provozu nelze použít. Prozatím není jisté, jestli je chování takovéto sítě ergodické, stacionární, normalizované nebo obojí. Konečný důkaz o tom stále neexistuje. Důraz musí být také kladen na jeho stochastický proces a životní cyklus.

2 Formulace cílů řešení

Hlavním cílem dizertační práce je návrh nové metody detekčních mechanizmů v oblasti konvergovaných sítí. Cílem je využít současných poznatků možností vyhodnocení chování a výskytu anomálií provozu v takovýchto sítích a navrhnout řešení nová.

Součástí vytčených cílů je návrh a implementace algoritmu pro analýzu a detekci dat v konvergovaných sítích. Tento algoritmus či skupinu algoritmů modelovat a otestovat v některém z programovacích nebo skriptovacích jazyků. Jmenovitě jsou jednotlivé cíle následující:

1. Analýza současného stavu algoritmů a prostředků k detekci provozu.
2. Návrh nové metody detekce anomálií provozu.
3. Vývoj algoritmu či skupiny algoritmů vycházejících z algoritmů evolučních.
4. Model, který bude ověřovat funkčnost těchto algoritmů.
5. Publikační činnost.

2.1 Specifikace výzkumu

Jelikož je dané téma vlastní práce velice obsahově široké, bylo nutné se zaměřit na specifický předmět výzkumu.

1. Konvergované sítě obecně představují rychle se rozvíjející oblast. Pro vlastní účely práce byl zvolen princip sběru dat pomocí protokolu NetFlow a to také s ohledem na možnosti a vybavení laboratoře, kde byl výzkum prováděn.
2. Tématika bezpečnosti sítí stále narůstá na významu a konvergované sítě jsou nedílnou součástí této tématiky. Do této tématiky jsou zařazeny problematiky predikce, detekce a obrany. Z pohledu bezpečnosti sítí byla zvolena problematika detekce, která je základním předpokladem pro řešení predikce a obrany.
3. Z oblasti bezpečnostní detekce byla zvolena tématická sekce detekce anomálií. Ač anomálie mohou představovat jakékoliv tendence, odchylky či směry, práce se soustředí na možnosti detekce anomálií jako takových a možnosti nasazení evolučních algoritmů.
4. Z pohledu neprobádaných směrů je práce soustředěna na životnost a životní cyklus anomálního jevu. Pro snadnější identifikaci a prezentaci výsledků bylo zvoleno téma botnetových sítí. Toto téma je ovšem také velice obsáhlé, proto se práce soustředí na řídicí a kontrolní zprávy šířené mezi napadenými stanicemi a řídicími servery.

2.2 Přehled týkající se literatury

Mimo publikace (články, knihy a konferenční příspěvky) průběžně citované v dizertační práci navázal daný výzkum především na následující publikace rozdělené na jednotlivé problematiky.

Síťová tomografie a její principy v datových sítích

V rámci dizertace byly zkoumány jednotlivé směry prací superpočítačového centra kalifornské univerzity CAIDA, úzce spolupracující s komunitou RIPE NNC. Techniky síťové tomografie jsou zde využity v díle „*Challenges in Inferring Internet Interdomain Congestion*“. Autoři zde navrhují řešení validace zahlcení sítě na základě metod síťové tomografie. Této části se také týká publikace autorů Zhang, Y., Ge, Z., Greenberg, A. a Roughan, M. „Network anomography“ [9].

Evoluční algoritmy

Existuje mnoho výborných publikací týkajících se obecně problematiky evolučních algoritmů. Významná část v práci byla čerpána z literatury „Genetické algoritmy a genetické programování“ od autora Josefa Hynka [15]. Dále byla práce inspirována knihou „Genetic Algorithms for Control and Signal Processing“ [16].

Principy detekcí provozu

Oblast detekcí provozu, přesněji řečeno detekcí anomálií provozu, má rozsáhlé spektrum možností aplikace. V této oblasti bylo zásadních prací hned několik. Jednak se týkaly samotné možnosti nasazení genetických algoritmů v oblasti detekce anomálií provozu a dále obecně možnostmi a principy detekcí anomálií provozu.

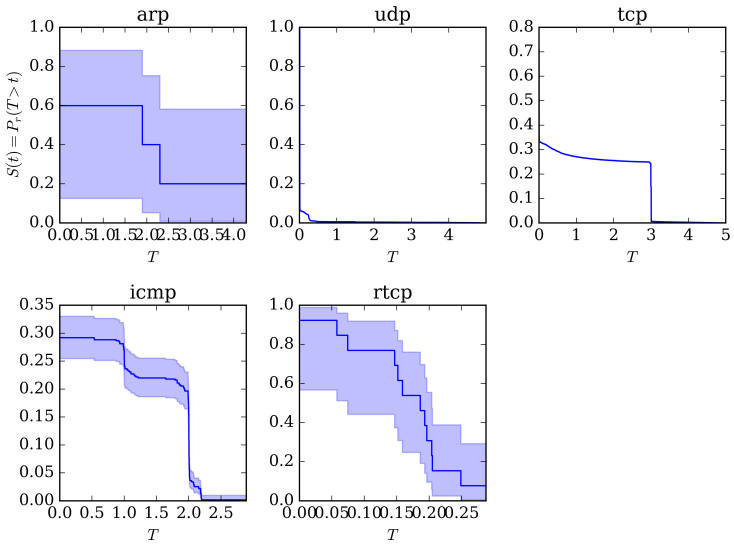
Pracemi týkající se tohoto tématu jsou: „An Entropy-Based Network Anomaly Detection Method“ autorů BEREZIŃSKI, Przemysław, Bartosz Jasiul a M. Szpyrka [4], „ASTUTE: Detecting a Different Class of Traffic Anomalies“ autorů Silveiray, F., Diot, C., Taft, N,; Govindan, R. [10]. Dále to jsou „Network Traffic Anomaly Detection“ [6], „A Genetic Algorithm For Solving the Eeuclidean Distance Matrices Completion Problem“ [8].

3 Hlavní výsledky

V rámci řešení práce a naplňování zvolených cílů bylo provedeno testování vlastních programů pro práci s algoritmy. Pro testování genetických algoritmů byly vytvořeny ilustrační úlohy v jazyce Python k ověření jejich teorie a následně byl navržen vlastní algoritmus pro behaviorální analýzu a vytvořen model síťové sondy, který se skládá z kolektoru NetFlow zpráv a supervizoru.

3.1 Testování hypotézy analýzy přežití

Pro účely ověření hypotézy, že pro každé síťové spojení je definován jiný stav přežití takového provozu a také, že je vyjádřen tento stav jinou křivkou přežití, byl postupně vyvíjen vlastní kolektor NetFlow zpráv. V tomto kolektoru je možné implementovat vlastní algoritmy. Výsledky testů byly publikovány v [A8] a [A11]. Byly provedeny dva testy pro modelování životního cyklu. Jeden inicializační a druhý podporující danou hypotézu. Na obrázku 2 jsou zobrazeny křivky přežití pro jednotlivé protokoly z testovaného datasetu.



Obr. 2: Zobrazení křivek přežití síťových protokolů

Tato analýza přežití může být aplikována na jakýkoliv časový proces, jako je například návštěvnost internetových stránek. Začátek trvání procesu je příchod nového návštěvníka na internetové stránky a koncem doby je jeho odchod. Jedním z cílů

analýzy přežití je extrakce modelů z dat, které přibližují rozložení doby životnosti. Těmito modely lze odhadnout čas, kdy dojde k události ke vztaženému objektu.

Funkce přežití $S(t)$ je definována dle vztahu (11). Tato funkce definuje pravděpodobnost, že v čase t ještě nenastal konec události nebo ekvivalentně, pravděpodobnost přežití alespoň do doby t.

$$S(t) = Pr(T > t)\,, \tag{11}$$

kde platí podmínky $0 \leq S(t) \leq 1$ a $S(t)$ je nezvyšující se funkcí t, protože kumulativní distribuční funkcí T je $F_T = 1 - S(t)$.

Pro odhad funkce přežití je používána metoda parciální věrohodnosti Cox nebo Kaplan-Meier analýza (12).

$$\hat{S}(t) = \prod_{j=1}^{k} \left(\frac{n_j - d_j}{n_j} \right) = \hat{S}(t) \left(1 - \frac{d_t}{n_t} \right), \tag{12}$$

kde d_j koresponduje s počtem událostí, případně s počtem ukončených událostí v čase j, zatímco n_j je vztaženo k počtu objektů, které jsou stále pozorované v čase j. Níže jsou uvedena vstupní data z testovaného datasetu.

	IP_SOURCE	IP_DEST	T	C
0	89.176.9.204	192.168.1.66	1049	1
...				
5	89.176.9.204	192.168.1.66	1041	1
6	192.168.1.54	192.168.1.255	1	1
7	192.168.1.46	192.168.1.255	1501	1
8	192.168.1.54	192.168.1.255	33009	1
11	192.168.1.54	192.168.1.66	106325	1
...				

Tyto data následně vstupují do Kaplan-Meier analýzy a výstupem jsou požadované křivky přežití a její numerické hodnoty.

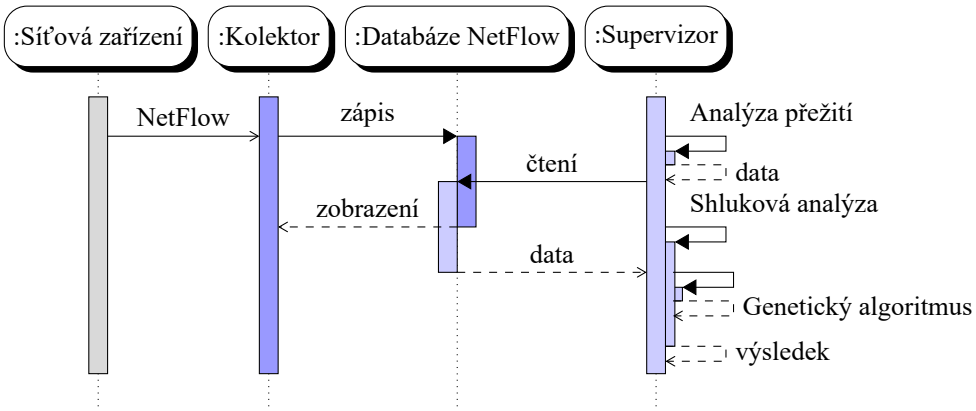
3.2 Návrh algoritmu a modelu

Vlastní model je založený na konceptu detekce anomálií provozu na základě statistické metody s využitím existujících funkcionalit síťových prvků NetFlow. Model využívá genetického algoritmu pro výpočty daných výsledků statistické metody analýzy přežití. Cílem tohoto modelu je schopnost analyzovat provoz v rozdílných časových úsecích a nalézt míru shody doby přežití pro jednotlivá síťová spojení. Takováto shoda dimenzí křivek nebo proměnných analýzy přežití potom představuje danou anomálii ve sledovaném provozu.

Vytvořená aplikace se skládá ze dvou hlavních modulů, a to modulu pro kolekci informací o provozu nazvaného kolektor (*collector*) a modulu pro vyhodnocení provozu nazvaného supervizor (*supervisor*). Modul kolektor naslouchá příchozí provoz a ukládá informace z NetFlow zpráv do interní databáze. Dále provádí základní analýzu provozu, konkrétně sumarizaci UNIX času pro jednotlivé komunikující IP adresy.

Modul supervizor je nezávislý na modulu kolektor. Pro každé unikátní spojení jsou z databáze NetFlow vyčteny informace o počtu spojení pro danou IP adresu, trvání každého spojení a o ukončení spojení. Následně jsou vypočteny hodnoty křivek přežití pro tato spojení a pomocí genetického algoritmu jsou data rozdělena do jednotlivých shluků ke zjištění, které provozy mají z pohledu podobnosti křivek přežití k sobě nejblíže.

Pro ověřování metod bylo nutné sestavit vlastní model síťové sondy. Celý funkční model je nazván GDP (Genetic Decision Probe). Základní princip je uveden na obrázku 3.



Obr. 3: Základní sekvenční schéma GDP

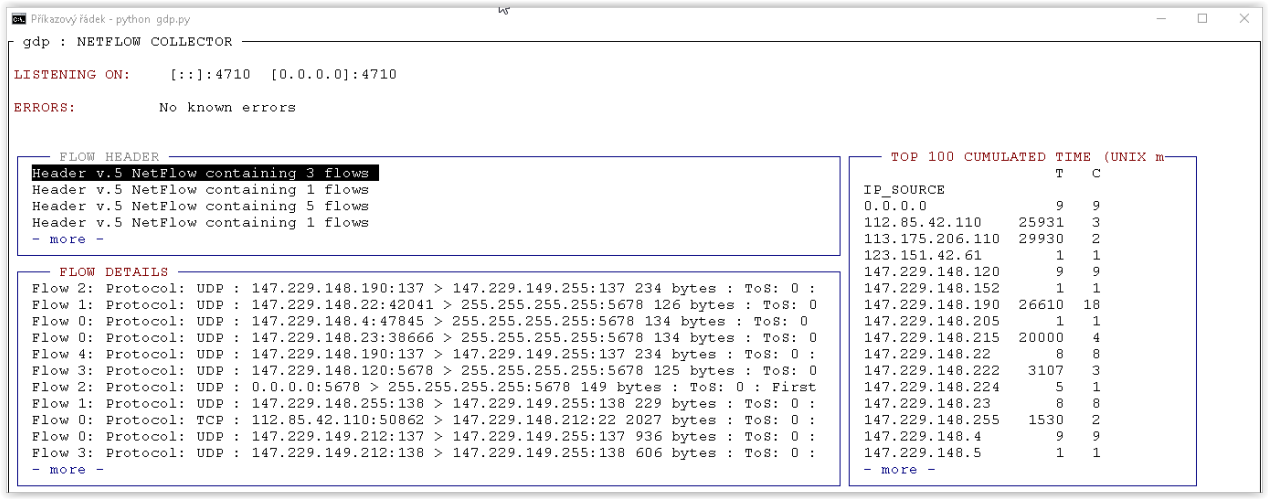
3.3 Modul kolektor

Koncept kolektoru je založen na schopnosti zpracovat NetFlow zprávy verze 1 a 5. Modul kolektor se skládá ze dvou hlavních částí. Hlavní spouštěcí program je definován v souboru `<gdp.py>`. Dalšími částmi jsou `<interfaces>`, `<core>`, `<database>` a `<common>`, které obsahují další operační soubory programu. Uživatelské rozhraní (*Terminal User Interface*) je vytvořeno za pomocí dostupného balíčku Python npyscreen a poskytuje vše potřebné pro interakci s uživatelem za použití systémové konzoly, viz obrázek 4.

Vytvořený program automaticky ukládá informace do databáze. Ta je tvořena souborem s názvem <dataset.sqlite3>. V konzoli jsou potom zobrazeny statistiky nejdéle trvajícího spojení a počet navázaných spojení. Tento kolektor byl prezentován v článku [A10]. Program lze stáhnout v uvedeném odkaze [18].

3.4 Modul supervizor

Modul supervizor načítá v průběžné smyčce data z databáze NetFlow kolektoru. Jsou využity principy síťové tomografie. Dle definice vztahu pro jednotlivá spojení SD (*Source to Destination*) jsou pro každé SD ze sondy P vypočteny hodnoty funkce



Obr. 4: Grafické zobrazení aplikace

přežití $S(t) = Pr(T > t)$ pro individuální časovou periodu, rovnice (13).

$$\hat{S}(t)_{SD} = \prod_{j=1}^{k} \left(\frac{n_j - d_j}{n_j} \right) \forall SD_P \,, \qquad S(t)_{IP} = \frac{\text{počet výskytů - počet ukončení}}{\text{počet výskytů}} \,, \tag{13}$$

kde čas výskytu události T je reprezentováno výpočtem z hodnot `(LAST - FIRST) + 1` z NetFlow toku SD a počet ukončení, resp. cenzorování je reprezentován volbou 0 nebo 1 databázovým dotazem `case when (LAST - FIRST)>0 then 1 else 0`.

Genová expresní matice je potom vyjádřena $GEM_{(m,n)} = m \times n$ pro časové okno $w(t)$, kde m představuje sadu řešení a n hodnoty $m = \{n1, n2, \ldots, n(t)\}$ sady m v daném lineárním prostoru. Příklad uveden níže.

```
192.168.1.66  1.00  0.73  0.64  0.55  0.45  0.45  0.27  0.18  0.18  0.18
192.168.1.255 1.00  0.42  0.42  0.28  0.28  0.14  0.14  0.07  0.00  0.00
89.176.9.204  1.00  1.00  0.66  0.33  0.32  0.00  0.00  0.00  0.00  0.00
...
```

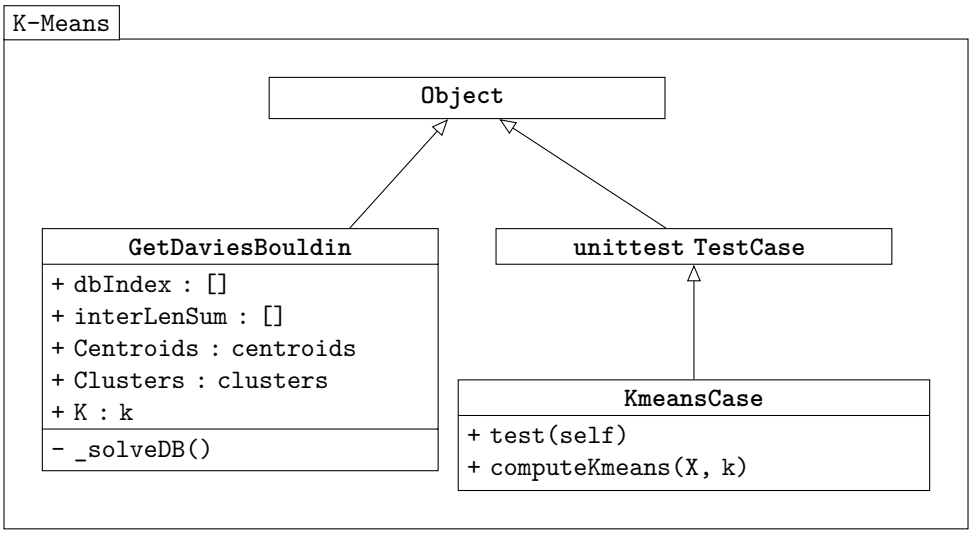
Získané hodnoty z analýzy přežití (viz výše) jsou podrobeny shlukové analýze pro účely zjištění podobnosti jednotlivých křivek přežití. Byl naprogramován algoritmus založený na principu K-průměrů (K-Means) a genetický algoritmus. Oba za účelem klastrování dat a k vzájemnému porovnání.

Pro účely testování vlastních algoritmů byla nejprve zvolena reálná testovací sada řidičů [19], kde byl sledován průměrný počet ujetých kilometrů v závislosti na průměrné rychlosti. Tato sada obsahuje 4 000 hodnot. Dále byl také zvolen testovací vzorek `make_circles` z balíčku `scipy python`.

K porovnání výsledků je použita časová hodnota běhu algoritmů a Daviesův-Bouldinův validační index. V případě genetického algoritmu je také zjišťována prů-

měrná hodnota Euklidovské vzdálenosti mezi prvky obsaženými ve shluku. Pro potřeby ohodnocení výstupů byla vytvořena třída `GetDaviesBouldin`, obrázek 5, vracející hodnotu DB indexu a vnitroshlukové vzdálenosti.

Algoritmus K-průměrů je sestaven podle probrané teoretické části. Je použit princip Lloydova algoritmu. Klíčovým krokem jsou průměry $x \in X$, které se rovnají argumentu $\min_{c \in \mathbb{R}^d} \sum_{x \in X} \|c - x\|^2$. Pracuje ovšem pouze se vzdáleností $\mathbf{d}(x, c) = \|x - c\|$. Pomalejší proces, ale odstraňující předešlé omezení, je zvolení centroidu c_i dle $\{x \in X | \phi_C(x) = c_i\}$. Třídní diagram K-průměrů je uvedena na obrázku 5.



Obr. 5: Třídní diagram K-Means.py

V prvním kroku jsou zvoleny náhodně $c_i = x_i \in X$. Dále je proveden výpočet Euklidovské vzdálenosti člena x_i vůči centroidu c_i a tento člen je vložen do nejvíce vyhovující skupiny. Centroidy jsou přepočítány a nahrazeny průměrnou hodnotou hodnosti členů shluku. Proces se opakuje dokud nejsou přiřazeni všichni členové.

V následujícím kroku konvergence algoritmus porovnává každého člena s vlastním shlukem a shlukem sousedním z pohledu Euklidovské vzdálenosti. V případě nalezení vhodnějšího shluku (klasteru) vybraného člena přesune do tohoto shluku. Tento krok se stále opakuje, dokud nenastane nulový počet přesunů členů.

Genetický algoritmus pracuje s chromozomy řešení daného problému. V tomto případě byl zvolen chromozom obsahující k genů pro k centroidů c_i shluků S. Chromozom, neboli jedinec α a následně populace P je v prvním kroku sestavena dle výrazu (14). Centroid $\mathbf{c}_i = \mathbf{x}_i$ a $\mathbf{x}_i$ představuje vektor hodnot $\mathbf{x}_i = \{n_1, n_2, \dots, n_M\}$, kde n představuje prvek sady M z genové matice.

$$\begin{aligned} \alpha_i = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k) \in C | \mathbf{c}_k = \mathbf{c}_i \in X\,, \\ P = \{\alpha_1, \alpha_2, \dots, \alpha_S\} \end{aligned} \tag{14}$$

Chromozom udržuje jen centroidy shluků a při volání ohodnocení jedince jsou přidruženy k těmto centroidům hodnoty $\mathbf{x}_i$. Pro ohodnocení jedince je využito Eukli-

dovské vzdálenosti a Daviesův-Bouldinův validační index. Genetický algoritmus minimalizuje sumu Euklidovské vzdálenosti pro všechny jednotlivé shluky. Výraz (15) představuje účelovou funkci daného problému.

$$\text{minimalizace} \sum_{k=1}^{k} \sum_{\mathbf{x}_n \in S_k} \|\mathbf{x}_n - \mathbf{c}_k\|^2, \quad \text{vzhledem k: } \mathbf{c}_k, S_k \tag{15}$$

Kriteriální funkce, tj. ohodnocení jedince (chromozomu) je prováděno na základě sumy průměrných vzdáleností uvnitř shluků, tj. $f(\alpha) \wedge fit_1(\alpha)$ a $\forall \alpha_i \in P : 0 < f(\alpha)_1 < f(\alpha)_2$. Potom genetický algoritmus volí jedince s nejnižší hodnotou sumy průměrných vzdáleností v populaci P. Pro zjednodušení běhu algoritmu je volána tato metoda z importované vlastní třídy `GetDaviesBouldin` z předchozího řešení K-průměrů. Celá koncepce kódu genetického algoritmu je uvedena na obrázku 6. Tato třída zajišťuje i výpočet Daviesův-Bouldinova validačního indexu a vrací její hodnotu v proměnné `dbIndex`. Tento validační index představuje druhou kriteriální funkci.

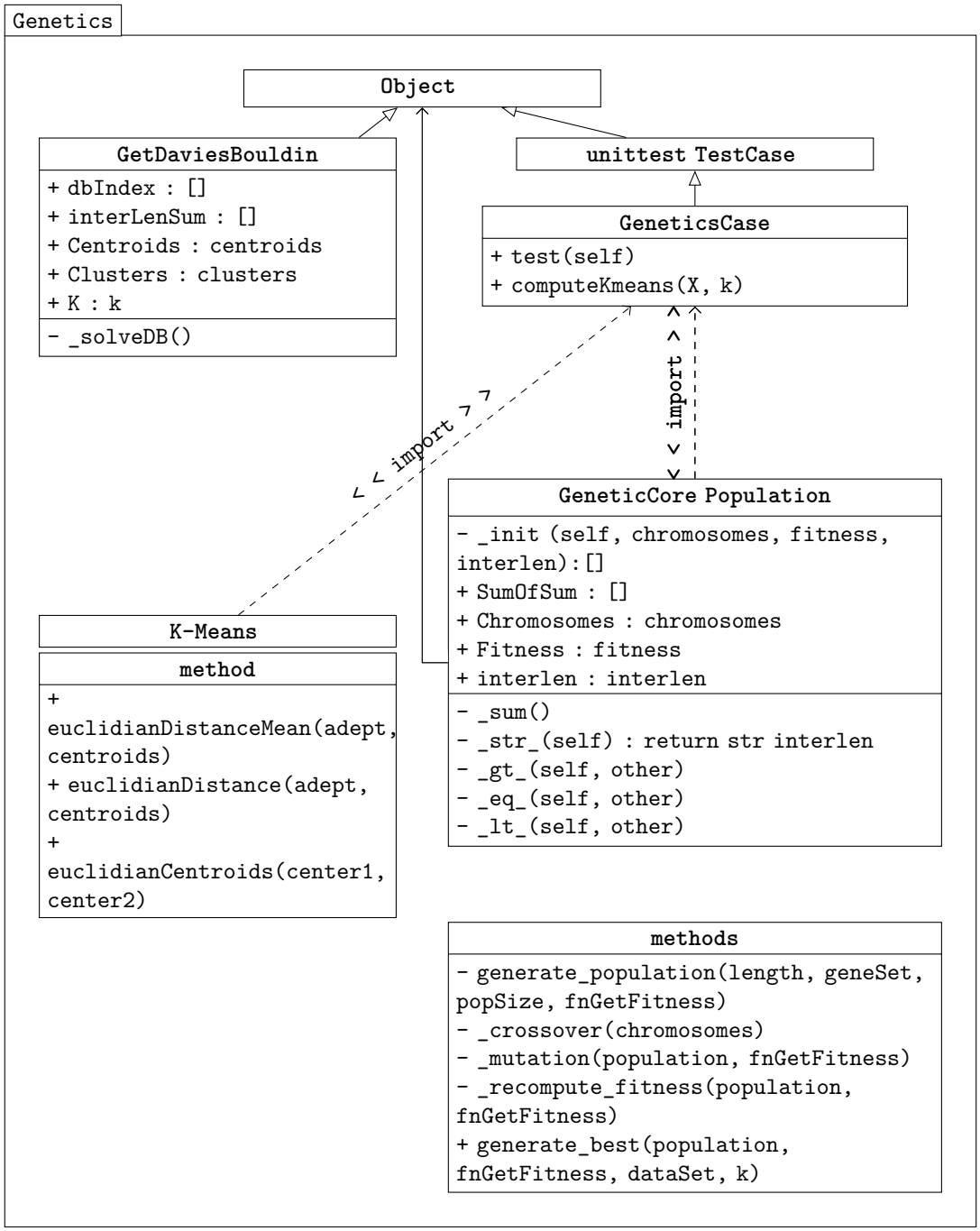
$$fit_2(\alpha) = DB_\alpha = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left\{ \frac{S_n(Q_i) + S_n(Q_j)}{S_n(Q_i, Q_j)} \right\}, \quad \forall \alpha_i \in P : 0 < f(\alpha)_1 < f(\alpha)_2 \tag{16}$$

Třída `Population` udržuje aktuální hodnoty populace `self.Chromosomes` a ohodnocení jedinců v `self.Fitness` a `self.interlen`. Obsahuje vlastní metody pro porovnání populací `OBJEKT.__lt__(self, other)` a dále `eq` a `gt`.

Vždy mezi dvěma populacemi je tedy hledána hodnota minima vnitro-shlukové vzdálenosti. Jedna populace představuje původní populaci a druhá populace představuje upravenou o křížení a mutaci s přepočítanými hodnotami kriteriálních funkcí.

Je tedy prováděno ohodnocení celých populací v rámci provedení elitismu. A to z toho důvodu, že je prováděno křížení jak vertikální, tak také horizontální mezi jednotlivými jedinci v populacích. Chybné výsledky (inf, null) po dělení nulou a sama sebou jsou filtrovány pomocí metody z balíčku numpy `np.ma.masked_invalid`. Tím je zajištěno, že nebudou vybráni chybní jedinci. V navrženém řešení je vytvořena metoda křížení na základě náhodného výběru jak ve vlastním chromozomu jednotlivých genů, tak také mezi chromozomy.

Mutace daného řešení zajišťuje, aby jednak nedošlo uvíznutí v lokálním a globálním minimu. Dále jsou také pomocí této mutace náhodně nahrazeny stávající hodnoty centroidů, protože v prvním kroku algoritmu je centroid $\mathbf{c}_i$ roven $\mathbf{x}_i$. V této



Obr. 6: Částečný diagram GeneticsCore.py

části prováděné mutace je využito třídy `GetDaviesBouldin`, která navrací průměr hodnot $\mathbf{x}_i \in S(k)$. Potom je chromozom tvořen (17).

$$\begin{aligned} \alpha_{i:\text{nové}} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k) \in C | \mathbf{c}_k = \emptyset S(k)\,, \\ P = \{\alpha_1, \alpha_2, \dots, \alpha_S\}\,, \\ \mathbf{c}_{k:\text{nové}} = \mathbf{c}_k = \mathbf{x}_i | i, k : \Omega \longrightarrow (1, N) \end{aligned} \tag{17}$$

Výsledkem testu je schopnost algoritmu porovnat křivky přežití jednotlivých ko-

munikujících uzlů a sdružit je podle podobnosti provozu křivek přežití a míry rizik. Níže je uvedeno složení vlastního datasetu v `DataFrame`.

```
              IP  a  b     c     d     e  f     g     h     i     j
0    89.176.9.204  0  1  1033  1041  1049  0  0.45  0.78  1.28  2.28
1    192.168.1.54  0  1  1501  1509  1553  0  0.37  0.44  0.51  0.58
2  79.143.185.229  0  1     0     0     0  0  0.00  0.00  0.00  0.00
3    192.168.1.46  0  1  1497  1501  1509  0  0.50  0.75  1.09  1.59
4    192.168.1.66  0  1  1033  1041  1049  0  0.79  0.88  0.98  1.09
```

Tento dataset tvoří genovou matici pro každý SD. Prvních pět hodnot tvoří časovou osu analýzy přežití a následných pět hodnot vlastní hodnoty časové osy. Tento nejjednodušší příklad ověřuje možnosti práce s výstupními hodnotami analýzy přežití a rozlišením jednotlivých křivek přežití.

Pomocí genetického algoritmu jsou zařazeny jednotlivé provozy do shluků. Tyto shluky představují křivky přežití, jejichž rozdíl vlastních hodnot ve vektorech vůči hodnotám centroidu se blíží z pohledu Euklidovské vzdálenosti k nule. Opět byly porovnány výsledky obou algoritmů, jak GA tak K-průměrů.

```
K-Means, 2 cykly do konvergence, (78ms)
DB Index 0.0016132173486533587
Průměrná vzdálenost uvnitř shluku: 0.9918338445433054
Blízká hodnota křivky, shluk 0:
79.143.185.229
Blízká hodnota křivky, shluk 1:
192.168.1.66
89.176.9.204
Blízká hodnota křivky, shluk 2:
192.168.1.54
192.168.1.46
```

Shluk číslo 1 obsahuje dvě adresy, z nichž každá představuje komunikaci mezi dvěma počítači s aplikací µTorrent. Další dvě ve shluku číslo 2 zastupují vnitřní IP adresy za směrovačem. Z tohoto pohledu algoritmus K-means přiřadil sobě podobné průběhy do stejného shluku. V dalším kroku byl ověřen genetický algoritmus. Ve výsledném výstupu je zobrazena průměrná vzdálenost uvnitř shluku pro každý cyklus.

```
GA, 4 cykly, populace = 4 (142ms)
Vzdálenost 1. cyklus : [155.33340769869454]
Vzdálenost 2. cyklus : [4.218783667088422]
Vzdálenost 3. cyklus : [2.211733592974675]
Vzdálenost 4. cyklus : [0.9918338445433054]

DB Index 0.0016132173486533587
Průměrná vzdálenost uvnitř shluku: 0.99183384454330537
Blízká hodnota křivky, shluk 0:
79.143.185.229
Blízká hodnota křivky, shluk 1:
89.176.9.204
192.168.1.66
Blízká hodnota křivky, shluk 2:
192.168.1.54
192.168.1.46
```

Z časového pohledu byl výpočet příkladu s reálným provozem pomocí GA náročnější, než výpočet pomocí algoritmu K-průměrů. Ve výsledku profilace algoritmu

je časově náročné vytváření nových instancí tříd `Population`. Při opakovaném pokusu byla objevena chybovost algoritmu, a to konkrétně přiřazení stejné IP adresy do dvou rozdílných shluků. Tato chyba nastávala v případě, že algoritmus ukončil výpočet s výsledkem hodnoty DB indexu $DB > 0,01$.

Navržený genetický algoritmus pracuje pouze s centroidy jako individuem a neudržuje data ve shluku, jako je tomu v případě K-průměrů. Při vytvoření populace pomocí volání metody `random.choice()` dochází při malém počtu chromozomu k paradoxu výběru shodných centroidů. Genetický algoritmus není poté schopný danou chybu napravit. Tento paradox je možné ovlivnit zvýšením počtu chromozomů v populaci, případně zvýšit hodnotu mutace. Od počtu deseti chromozomů v populaci již k chybě nedocházelo.

3.5 Diskuze a shrnutí výsledků

Z pohledu síťové tomografie byla detekce podobnosti životního cyklu provozu zvolena s ohledem na možnost detekovat takové cykly, které jsou generovány v jiných časových úsecích a v rozdílných sítích. Jedná se například o možnost detekce komunikace ransomware, který periodicky generuje TCP spojení. V tomto případě je možné se zaměřit na životní cyklus a jako podklad pro další zkoumání a rozhodovací proces získat přehled komunikujících uzlů s podobným průběhem, ať se již jedná například o globálně zachycený provoz napříč sítěmi. Dále je již možné se zaměřit pouze na vybraná spojení a určit, zda se jedná o nevyžádaný provoz.

Původně zamýšlené využití programu pro simulaci sítí OMNeT++ se ukázalo být velmi složité z pohledu programování a časové náročnosti. Oproti tomu Python poskytuje dostačující podporu a potřebné knihovny pro výzkum a vývoj.

Genetické algoritmy mohou najít mnohé uplatnění. Velmi důležité je zvolení koncepce ověřování výsledků – ohodnocující funkce. Ta vždy musí odpovídat danému problému. Je také vhodné určit, zda má smysl použít genetický algoritmus nebo nikoli. Genetické algoritmy patří mezi heuristické metody řešení daného problému. Jsou vhodné zejména tam, kde není možné jinou metodou či algoritmem v rozumném čase dospět k výsledku řešení nebo není známá funkce daného řešení. Jedná se zejména o NP těžké úlohy, na které se dají převést všechna ostatní řešení. V mnoha úlohách mohou genetické algoritmy selhávat a nepředstavují univerzální nástroj.

Původně uvažované řešení zjištění vzdáleností křivek za pomocí minimalizace kostry grafu síťových spojení se ukázalo být neefektivní v tom směru, že by samotný výstup obsahoval pouze jeden shluk nejkratších vzdáleností.

Navržený genetický algoritmus částečně využívá principu algoritmu K-průměrů, ale pracuje pouze se samotnými centroidy, které představují chromozom řešení. Následně aktualizuje shluky a centroidy na základě Euklidovské vzdálenosti. V paměti nejsou udržovány jednotlivé shluky. Algoritmus minimalizuje hodnotu Davies-Bouldinova validačního indexu a hodnotu sumy Euklidovských vzdáleností všech shluků.

Samotné řešení poskytuje práci s pevně stanoveným počtem shluků. Uvedený algoritmus je vhodné rozšířit o možnost dynamicky zvyšovat či snižovat počet těchto shluků. Dále také o možnost nastavit rozpětí Euklidovské vzdálenosti.

4 Závěr a využití

V rámci dizertační práce byl navržen a implementován nový prvek – sonda síťových anomálií. Dále byly vypracovány vlastní algoritmy pro behaviorální analýzu, které využívají genetické algoritmy. Cíle definované v kapitole 2 se podařilo během řešení naplnit.

Navržený model síťového analyzátoru a vlastní implementace algoritmů poskytují pohled na životní cyklus datového provozu a provádí analýzu podobnosti provozů. Samotná definice „anomálie“ je velmi široký pojem a je těžké určit co vlastní anomálii představuje. Například při útoku typu DoS se může jednat o dočasný zvýšený zájem o určitou službu, která vyvolá falešně pozitivní reakci. V tomto případě je možné se zaměřit na životní cyklus, a jako podklad pro rozhodovací proces získat přehled komunikujících uzlů s podobným průběhem komunikace. Proto byla v daném řešení anomálií provozu uvažována podobnost křivek přežití, které jsou výstupem analýzy přežití. Jednotlivé hodnoty jsou dále zpracovány genetickým algoritmem a následně je vyhodnoceno jejich seskupení a souvztažnost.

Prvním vytyčeným cílem bylo analyzovat současný stav algoritmů a prostředků k detekci provozu. Byl probrán teoretický podklad a jejich současné využití. Následně bylo provedeno praktické měření jednak analyzátorem ENDACE PROBE EP7010-PS-FC, dále také detailněji zkoumán princip fungování NetFlow. Byla naprogramována vlastní sonda založená na měření RTT sbírající data ze sondy RIPE NNC. Testován byl také zapůjčený IDS systém GAiA společnosti Check Point Software Technologies Ltd.

V rámci analýzy současného stavu byla naprogramována první část modelu, a to kolektor NetFlow nazvaný GDP. Byl použit jazyk Python ve verzi 3. Správná funkčnost byla testována v zapojení se směrovači v programu GNS3. Dále byly použity laboratorní přepínače k zasílání NetFlow verze 1 a 5. Byl proveden sběr dat z veřejné a laboratorní sítě.

Druhým vytyčeným cílem bylo navrhnout novou metodu detekce anomálií provozu. Byl proveden úvodní srovnávací test programu OMNeT++ a Python a následně test na základě hypotézy využití analýzy přežití pro vlastní rozlišení provozu. V tomto testování bylo využito torrent klientů, vlastního botnet serveru a také dataset provozu z ČVUT. Z výsledků bylo usouzeno, že je teoreticky možné vytvořit vzory provozu na základě křivek analýzy přežití a sledovat jejich životní cyklus.

Další stanovený cíl a následující se již vzájemně doplňují. V rámci naplnění cíle vývoje algoritmu, či skupiny algoritmů vycházejících z algoritmů evolučních bylo provedeno praktické prozkoumání funkčnosti a vlivu operátorů genetických algoritmů. Byl navržen a naprogramován vlastní genetický algoritmus v jazyce Python.

V rámci plnění hlavního cíle byl dokončen návrh druhého modulu síťové sondy s názvem supervizor. Byl zde implementován genetický algoritmus, který je určen pro rozdělení zachyceného provozu převedeného do křivek přežití do jednotlivých shluků na základě Euklidovské vzdálenosti. Jedná se tedy o porovnání bodů všech průběhů, vektorů jednotlivých spojení.

V dalším možném vylepšení je vhodné provést paralelizace genetického algoritmu a pracovat na efektivnosti programového kódu. Samotný genetický algoritmus je připraven na práci s vektory libovolné délky. Není tedy omezen na zpracování pouze výsledků analýzy přežití. Tato představovala jedno z možných vyjádření vzorů provozu. Provedenými testy se potvrdilo, že pomocí navrženého algoritmu lze ve zvoleném časovém úseku rozlišit a párovat datové provozy podle jejich průběhu.

Na provedený výzkum by mohla navázat implementace algoritmů do programovatelných hradlových polí FPGA síťových karet. Pokračující projekty zaměřující se na implementaci FPGA dále navazují na vlastní řešené téma. Jedná se o jeden ze tří autorem vypsaných projektů, a to o možnost detekce dat v xPON sítích mezi koncovými a řídícími jednotkami. Během řešení dizertační práce byl rozpracován návrh nasazení genetického algoritmu a proveden návrh a testování pseudonáhodných generátorů. Generátory jako takové hrají důležitou roli a jsou genetickými algoritmy hojně využívány. Konkrétně se jednalo o návrh ASGG a GEFFE pseudonáhodného generátoru.

V oblasti detekce anomálií by bylo zajímavé zaměřit další výzkum na metody umělého imunitního systému, které v současné době představují vhodného nástupce neuronových sítí a genetických algoritmů a kombinují jejich možnosti. Zajímané využití v konvergovaných sítích a jednotlivých aplikacích by mohly nalézt soutěživé algoritmy.

Dílčí výsledky byly průběžně publikovány a všechny vytčené cíle disertační práce byly dosaženy.

Literatura

[1] FERNANDEZ, Eduardo B. *Security patterns in practice: designing secure architectures using software patterns*. United Kingdom: John Wiley & Sons, Ltd., 2013, xxi, s. 558. Wiley series in software design patterns. ISBN 978-1-119-99894-5.

[2] Introduction to Cisco IOS NetFlow - A Technical Overview. CISCO SYSTEMS, INC. *CISCO* [online]. 2012 [cit. 2016-01-22]. Dostupné z: </http://www.cisco.com/NetFlow/>.

[3] VARDI, Y. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. *Journal of the American Statistical Association*. 1996, vol. 91, v. 433. DOI: 10.2307/2291416.

[4] BEREZIŃSKI, Przemysław, Bartosz JASIUL a Marcin SZPYRKA. An Entropy-Based Network Anomaly Detection Method. *Entropy* [online]. 2015, **17**(4), 2367-2408 [cit. 2017-04-08]. DOI: 10.3390/e17042367. ISSN 1099-4300. Dostupné z: <http://www.mdpi.com/1099-4300/17/4/2367/>.

[5] SMITH, Lindsay I. *A tutorial on Principal Components Analysis* [online]. 2002, [cit. 2017-04-10]. Dostupné z: <http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf>.

[6] HUANG, Hong, Hussein AL-AZZAWI a Hajar BRANI. Network Traffic Anomaly Detection. *Semantic Scholar* [online]. Allen Institute for Artificial Intelligence, 2014, , 26 [cit. 2017-04-11]. arXiv:1402.0856v1 [cs.CR]. Dostupné z: <https://pdfs.semanticscholar.org/2ad0/8da69a014691ae76cf7f53534b40b412c0e4.pdf>.

[7] SOMVANSHI, Divya a R.D.S. YADAVA. Boosting Principal Component Analysis by Genetic Algorithm. *Defence Science Journal* [online]. 2010, **4**(60), 7 [cit. 2017-04-12]. DOI: 10.1.1.902.7675. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.902.7675&rep=rep1&type=pdf>.

[8] RIVERA-GALLEGO, Wilson. A Genetic Algorithm for Solving the Euclidean Distance Matrices Completion Problem. *SAC* [online]. 1998, , 5 [cit. 2017-04-12]. ACM l-S81 13-086-4199/000. Dostupné z: <http://slapper.apam.columbia.edu/bib/papers/river_b_99.pdff>.

[9] ZHANG, Y., Ge, Z., Greenberg, A. a Roughan, M. Network anomography. *(2005) Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC,* s. 317-330. [cit. 2017-04-11] <http://conferences.sigcomm.org/imc/2005/papers/imc05efiles/zhang/zhang.pdf>.

[10] SILVEIRAY, F.; Diot, C.; Taft, N.; Govindan, R.: ASTUTE: Detecting a Different Class of Traffic Anomalies. In SIGCOMM Proceedings, New Delhi, India, Září 2010 [cit. 2017-04-03], [online]. DOI: http://doi.acm.org/10.1145/1851275.1851215. Dostupné z: <http://www.sigcomm.org/ccr/papers/2010/October/1851275.1851215>.

[11] *Stratosphere IPS Project* [online]. 2015 [cit. 2017-04-12]. Dostupné z: <https://stratosphereips.org/>.

[12] Plixer: Flow Analytics. PLIXER INTERNATIONAL, INC. *Plixer - Malware Incident Response* [online]. 2016 [cit. 2016-01-20]. Dostupné z: <https://www.plixer.com/Scrutinizer-Netflow-Sflow/flow-analytics.html>.

[13] *PRTG Network Monitor - Powerful Network Monitoring Software* [online]. Nuremberg: Paessler, 2017 [cit. 2017-04-04]. Dostupné z: <https://www.paessler.com>.

[14] MCDONNELL, John R, Robert G REYNOLDS a David B FOGEL. *Evolutionary programming IV: proceedings of the Fourth Annual Conference on Evolutionary Programming*. Cambridge, Mass.: MIT Press, c1995, xx, 805 p. ISBN 02-621-3317-2.

[15] HYNEK, Josef. *Genetické algoritmy a genetické programování*. 1. vyd. Praha: Grada, 2008, 182 s. ISBN 978-80-247-2695-3.

[16] MAN, Kim F., Kit S. TANG, Sam KWONG a Wolfgang A. HALANG. *Genetic algorithms for control and signal processing*. 1. S.l.: Springer, 1997. ISBN 978-144-7112-419.

[17] S.C.S. Silva, R.M.P. Silva, R.C.G. Pinto, a R.M. Salles, ”Botnets: A survey,”Computer Networks, vol. 57, s. 378–403, 2013.

[18] NSR – Network Security Research. 2014–2016, [cit. 2017-04-10]. Dostupné z: <http://http://nsr.utko.feec.vutbr.cz/software.php>.

[19] “Introduction to K-means Clustering”, DATASCIENCE, 2017. [cit. 2017-06-12], [online]. Dostupné z: <https://www.datascience.com/blog/.

Publikační činnost

[A1] OUJEZSKÝ, V.; ŠKORPIL, V.; JURČÍK, M. Network Tomography Overview and Botnet Network Estimation, Part I. Access Server, 2015, roč. 13, č. 6, s. 1-4. ISSN: 1214-9675.

[A2] POLÍVKA, M.; OUJEZSKÝ, V.; ŠKORPIL, V.; Modem Network Vulnerabilities and Security Testing – Actual Threats, TSP2015

[A3] OUJEZSKÝ, V.; NOVOTNÝ, B.; Reliability and Availability Calculation for the Educational Laboratory. In Proceedings of the 21st Conference STUDENT EEICT 2015. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. s. 581-588. ISBN: 978-80-214-5148-3.

[A4] OUJEZSKÝ, V.; ŠKORPIL, V. Data Field Transformation from Ethernet Frame. International Journal of Emerging Research in Management and Technology, 2014, roč. 3, č. 4, s. 4-8. ISSN: 2278-9359.

[A5] OUJEZSKÝ, V.; ŠKORPIL, V. Cryptographic Sequence Generators for Stream Cipher and Their Behavioral Description. International Journal of Advanced Research in, Computer Science and Software Engineering, 2014, roč. 4, č. 3, s. 106-121. ISSN: 2277-128X.

Článek byl citován v:

- Volodymyr Maksymovych. *Poisson pulse sequence generators based upon modified Geffe generators*, Polytechnika Krakowska, DOI: 10.4467/2353737XCT.14.054.3962.

[A6] ŠKORPIL, V.; OUJEZSKÝ, V. Služby telekomunikačních sítí. Brno: VUT Brno, 2014. s. 1-130.

[A7] OUJEZSKÝ, V. Videokonferenční technologie v praxi – Vidět a slyšet na dálku. Upgrade IT!, 2008, roč. IV., č. 1/ 2008, s. 44-45. ISSN: 1801-5026.

[A8] OUJEZSKÝ, V.; HORVÁTH, T.; ŠKORPIL, V. Modeling Botnet C&C Traffic Lifespans from NetFlow Using Survival Analysis. In Proceedings of the 39th International Conference on Telecommunication and Signal Processing, TSP 2016. International Conference on Telecommunications and Signal Processing (TSP). Vienna, Austria: 2016. s. 50-55. ISBN: 978-1-5090-1287-9. ISSN: 1805-5435.

[A9] OUJEZSKÝ, V.; HORVÁTH, T. Case Study and Comparison of SimPy 3 and OMNeT++ Simulation. In Proceedings of the 39th International Conference on Telecommunication and Signal Processing, TSP 2016. International Conference on Telecommunications and Signal Processing (TSP). Vi-

enna, Austria: 2016. s. 15-19. ISBN: 978-1-5090-1287-9. ISSN: 1805-5435.

[A10] OUJEZSKÝ, V.; HORVÁTH, T. NetFlow Console Collector–-Analyzer Developed in Python Language. In International Interdisciplinary PhD Workshop 2016. Brno: Brno University of Technology, Antonínská 548/1, Brno 601 90, 2016. s. 107-110. ISBN: 978-8-0214-5387-6.

[A11] OUJEZSKÝ, V.; HORVÁTH, T.; ŠKORPIL, V. Botnet C&C Traffic and Flow Lifespans Using Survival Analysis. International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, 2017, roč. 6, č. 1, s. 38-44. ISSN: 1805-5443.

[A12] HORVÁTH, T.; OUJEZSKÝ, V.; MÜNSTER, P.; VOJTĚCH, J.; HAVLIŠ, O.; SIKORA, P. Modified GIANT Dynamic Bandwidth Allocation Algorithm of NG- PON. Journal of Communications Software and Systems, 2017, roč. 13, č. 1, s. 15-22. ISSN: 1845-6421.

[A13] OUJEZSKÝ, V.; HORVÁTH, T. Traffic Analysis Using NetFlow and Python. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, 2017, ISSN: 2391-6761.

[A14] OUJEZSKÝ, V.; HORVÁTH, T.; ŠKORPIL, V. Traffic Similarity Observation by Using a Genetic Algorithm. 2017. Toho času odevzdáno k recenzi.

Profesní životopis

Zaměstnání

2016–doposud IBM Client Inovation Center Central Europe
2017–doposud Vysoké učení technické v Brně
2013–2016 T-Mobile Czech Republic a.s.
2006–2013 T-System Czech Republic a.s.

Studium

2009–2011 Vysoké učení technické v Brně – obor Telekomunikační a informační technika
2006–2009 Vysoké učení technické v Brně – obor Teleinformatika

Účast na projektech

2017 VI2VS/428 Detekce bezpečnostních hrozeb na aktivních prvcích kritických infrastruktur, MV ČR
2017 VI2VS/422 Redukce bezpečnostních hrozeb v optických sítích, MV ČR

Školní a pedagogické aktivity

2013–2017 Správa laboratoře transportních sítí centra SIX
2017 Člen IEEE
2014 Recenze IEEE Manuscript TCAD
2014 Oponentura diplomové práce
2015 Vedení diplomové práce

Certifikace, testy

CCNA Cisco Certified Network Associated
CCNP Cisco Certified Network Professional

Školení

Cisco ICDN1, ICDN2, A1, A2, R2, S1
HP & ComWare Configuration
Business Management (AKAD/IMAKA) (23440/92-34)
Linux LXI2-20120200015, LXI3 - 20130200011
SkillSoft Cisco Route, VHDL

Abstrakt

Tomografie síťového provozu představuje dnes již nedílnou součást v oblasti konvergovaných sítí a systémů k detekci jejich behaviorálních vlastností. Dizertační práce se zabývá výzkumem její implementace s využitím evolučních algoritmů. Výzkum byl zejména soustředěn na inovaci a řešení behaviorální detekce toků dat v sítích a jejich anomálií s využitím síťové tomografie a evolučních algoritmů. V rámci řešení dizertační práce byl navržen nový algoritmus, vycházející ze základů statistické metody analýzy přežití v kombinaci s algoritmem genetickým. Navržený algoritmus byl testován ve vlastním vytvořeném modelu síťové sondy za pomocí programovacího jazyka Python a laboratorních síťových zařízení Cisco. Provedené testy prokázaly základní funkčnost navrženého řešení.

Abstract

Nowadays, the traffic tomography represents an integral component in converged networks and systems for detecting their behavioral characteristics. The dissertation deals with research of its implementation with the use of evolutionary algorithms. The research was mainly focused on innovation and solving behavioral detection data flows in networks and network anomalies using tomography and evolutionary algorithms. Within the dissertation has been proposed a new algorithm, emerging from the basics of the statistical method survival analysis, combined with a genetics' algorithm. The proposed algorithm was tested in a model of a self-created network probe using the Python programming language and Cisco laboratory network devices. Performed tests have shown the basic functionality of the proposed solution.