VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

DETEKCE ANOMÁLIÍ VE WI-FI KOMUNIKACI

WI-FI COMMUNICATION ANOMALY DETECTION

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE **ZBYNĚK LIČKA**
AUTHOR

VEDOUCÍ PRÁCE **Mgr. Ing. PAVEL OČENÁŠEK, Ph.D.**
SUPERVISOR

BRNO 2022

Abstrakt

Práce se zabývá detekcí anomálií při komunikaci pomocí technologie IEEE 802.11 (Wi-Fi) na linkové vrstě OSI. K detekci anomálií byla zvolena metoda neuronových sítí, konkrétně LSTM rekurzivních neuronových sítí. Na začátku je popsána oblast zaměření a motivace k detekci anomálií v prostředí počítačových sítí. Poté jsou popsány různé techniky detekce anomálií, které jsou v oblasti počítačových sítí běžně používány. Pokračuje analýza požadavků na systém umožňující detekci anomálií a volba vhodné metody pro tyto účely. Dále je popsán návrh a způsob implementace systému a zvoleného modelu. Poté pokračuje zpráva o testování, vyhodnocení experimentů a diskuze.

Abstract

This thesis deals with anomaly detection in communication using the IEEE 802.11 technology (Wi-Fi) at the data link layer of OSI. The neural network method, specifically LSTM recurrent neural network, has been chosen for anomaly detection purposes. Initially, the focus area and motivation for anomaly detection in a computer network environment is described. Then, various methods for anomaly detection in computer networking are described. Thesis continues with analysis of the requirements for the system and a draft of the final system, including the chosen method, continuing with implementation of the system and model. Testing and evaluation of results takes place before the theses' conclusion.

Klíčová slova

Umělá inteligence, IEEE 802.11, detekce anomálií, strojové učení, síťové zabezpečení, LSTM RNN, neuronové sítě

Keywords

Artificial intelligence, IEEE 802.11, anomaly detection, machine learning, network security, LSTM RNN, neural networks

Citace

LIČKA, Zbyněk. *Detekce anomálií ve Wi-Fi komunikaci*. Brno, 2022. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Mgr. Ing. Pavel Očenášek, Ph.D.

Detekce anomálií ve Wi-Fi komunikaci

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Mgr. Ing. Pavla Očenáška, Ph.D.. Další informace mi poskytl Ing. Petr Chmelař. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Zbyněk Lička
27. července 2022

Poděkování

Rád bych poděkoval panu Mgr. Ing. Očenáškovi, Ph.D. a panu Ing. Chmelaři za jejich trpělivost a trvalou pomoc při tvorbě této práce.

Ústav informačních systémů (UIFS) Akademický rok 2021/2022

Zadání bakalářské práce

Student: **Lička Zbyněk**
Program: Informační technologie
Název: **Detekce anomálií ve Wi-Fi komunikaci**
Wi-Fi Communication Anomaly Detection
Kategorie: Bezpečnost

Zadání:

1. Seznamte se s principy analýzy anomálií v prostředí systémů počítačových sítí.
2. Analyzujte požadavky na systém umožňující analýzu provozu na nižších vrstvách OSI standardu IEEE 802.11 a detekci nestandardního chování připojených zařízení.
3. Navrhněte systém pro detekci anomálií dle instrukcí vedoucího práce, který bude založen na umělé inteligenci.
4. Navržený systém implementujte.
5. Implementovaný systém ověřte na vhodně zvolených reálných datech.
6. Diskutujte získané výsledky a možnosti dalšího rozšíření.

Literatura:

- Kurose, J. F. Computer networking: A top-down approach. Pearson, Essex, 2017, ISBN 978-1-292-15359-9.
- Stallings, W. Network security essentials: Applications and standards. Hoboken, 2016, ISBN 978-0-13-452733-8.
- Bishop, M. Computer security: Art & Science. Addison-Wesley, Boston, 2003, ISBN 0-201-44099-7.
- Buczak, A., Guven, E.. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications surveys and tutorials. IEEE, 2016, 18(2), s. 1153-1176.
- Kruegel, Ch., Vigna, G. Anomaly Detection of Web-based Attacks. In: Proceedings of the ACM Conference on Computer and Communications Security. ACM, Washington, DC, USA. 2003.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Očenášek Pavel, Mgr. Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 29. července 2022
Datum schválení: 11. října 2021

Obsah

Kapitola 1

Úvod

Internet se za posledních 30 let stal výraznou částí života téměř každého z nás. Jedná se o technologii přivedenou na svět ke sdílení znalostí s celým světem. Avšak ne každý používá tuto technologii k šlechetným účelům. V naší společnosti se objevují lidé, kteří se snaží využít chyb systémů ve svůj prospěch či k vlastnímu pobavení, nehledě na způsobené škody. Tato hrozba je do poslední chvíle často přehlížena, a následky bohužel mohou být drastické. Z tohoto důvodu začaly vznikat ochranné mechanismy, které brání sítě a počítače, před útočníky, tzv. hackery. Tyto mechanismy, zabraňující či odhalující útoky, kolektivně označujeme jako počítačové zabezpečení. Často se jedná o systémy, které nabízejí pouze částečnou ochranu a musí být kombinovány, aby bylo docíleno teoreticky plného zabezpečení. Tímto vzniká závod ve zbrojení mezi architekty bezpečnostních systémů a hackery. Nejvýraznější a neproblematičtější částí této situace je, že útočníkovi se stačí zaměřit na nejslabší článek systému. Takovým útokům je snadné zabránit, pokud jejich podobu známe. Nejnebezpečnější útoky jsou však často ty, které nebyly v minulosti dosud spatřeny, a tudíž je jejich forma neznámá. Od toho se odvíjí zaměření této práce, detekce anomálií ve Wi-Fi komunikaci. Jedním ze způsobů, jak detekovat nikdy předem neviděné útoky, je implementovat systém, který je schopen rozpoznávat normální chování od nenormálního. Tato práce se zabývá implementací právě takového systému. Využívá k tomu umělé inteligence, která nabízí řadu modelů schopných tuto úlohu splnit.

Následuje bližší popis oblasti detekce anomálií a technik umělé inteligence, které jsou pro účely detekce anomálií v rámci počítačových sítí používány. Dále je v nezbytném rozsahu popsán standard IEEE 802.11 a možné útoky na tento protokol. Po teoretické části následuje analýza požadavků na cílový systém umožňující detekci anomálií ve Wi-Fi komunikaci. Na tuto kapitolu přímo navazuje návrh a implementace systému. Pokračuje zpráva o testování a závěr.

Kapitola 2

Metody detekce anomálií pro strojové učení

Existuje celá řada metod pro detekci anomálií, značná část z nich založených na strojovém učení. V této kapitole lze nalézt základní úvod do oblasti detekce anomálií pomocí umělé inteligence a popis jednotlivých metod, v dnešní době běžně používaných, včetně výčtu jejich pozitivních a negativních vlastností. Každá z těchto metod disponuje určitými vlastnostmi, a z tohoto důvodu lze k řešení určité úlohy efektivně využít pouze část všech možných metod. Postup pro výběr vhodné metody pro řešení této práce bude dále popsán v kapitole 4.

Jedním ze zajímavých pohledů na strojové učení je, že daný algoritmus se ve skutečnosti snaží modelovat data, tj. nalézt funkci, která je na základě dostupných parametrů schopná vyvodit správnou třídu datového bodu. Ve skutečnosti tato funkce teoreticky existuje, avšak její podoba je pro většinu složitějších dat neznámá. Strojové učení je nástroj, kterým lze tuto funkci aproximovat.[10]

Jedním z kritérií, které je potřeba brát na zřetel, je interpretovatelnost výsledku. U tohoto kritéria můžeme dát do kontrastu např. asociační pravidla, které hledají jasná a lehce pochopitelná pravidla ve vstupních datech, a neuronové sítě aproximující model dat pomocí vážených vstupů a aktivační funkce. Zda je chování anomální by bylo vyjádřeno maticí vah, která je jen těžce interpretovatelná. Z tohoto důvodu je žádoucí navrhovat systém detekce způsobem, který dává prostor intepretovatelnosti výsledku detekce.

Výčet metod v této práci byl proveden na základě [7]. Popis jednotlivých metod byl převážně převzat z [3].

2.1 Terminologie

Před výčtem a porovnáním jednotlivých metod je třeba ustanovit řadu základních pojmů často užívaných v literatuře této oblasti.

2.1.1 Anomálie

Dle [4, str. 2] je *anomálie* vzor v datech, který nesouhlasí s očekávanou a dobře definovanou strukturou dat. Detekce anomálií je tedy proces hledání těchto vzorců v datové sadě. Pojem

anomálie je často v literatuře přímo zaměnován s pojemem *odlehlá hodnota* (dále anglicky *outlier*). Dalo by se diskutovat, zda jsou zaměnitelné.

Outlier [11, str. 1] popisuje outlier jakožto hodnotu, která může vzniknout chybou v experimentu, pozorování, či vyhodnocení nebo se může jednat o hodnotu ležící na pokraji běžné distribuce dat. V případě, že hodnota je pouhým extrémním případem běžné distribuce dat, měla by být považována za normální hodnotu, tj. ne anomální. Při porovnání těchto dvou pohledů lze říct, že anomálie je outlier, ale ne každý outlier je anomálie.

[4, str. 3] popisuje řadu úskalí, která se při řešení problému detekce anomálií mohou vyskytnout.

- Definování normální struktury dat je náročné. Obvzláště obtížné je najít hranici mezi outlier, kteří jsou normálními daty, a anomálními daty.
- Pokud jsou anomálie výsledkem zlomyslné činnosti, často se je jejich strůjci snaží zamaskovat svou aktivitu napodobením normálního chování a ztížit tak detekci. Tento bod je obzvláště aplikovatelný pro tuto práci, neboť je protokol IEEE 802.11 cílem mnoha rozličných útoků, jak lze vyčíst například z [15]. Pokus o zamaskování útoku a jeho následnou detekci lze najít např. v [19].
- Normální chování může být měnné v čase. Tento problém se prakticky řeší přetrénováním modelu po uplynutí vhodné doby. Změna normálního chování může být například důsledkem změny AP (Access Point, viz. kapitola 4) s podporou rozdílné verze protokolu IEEE 802.11.
- Odhadnout citlivost systému pro detekci je taktéž důležité. Již málé změny v struktuře dat mohou být anomální.
- Označené datové sety nemusí být dostupné.

Tento seznam je výtah problémů, které se mohou projevit při snaze o detekci anomálií v IEEE 802.11 komunikaci. V [4] je dále zmíněný datový šum, který může být podobný anomáliím a může tak stěžovat jejich detekci.

Motivace k detekci anomálií Aby vznikla iniciativa pro detekci anomálií, musí být anomálie pro výzkumníka zajímavé. Jednou z motivací k užití detekce anomálií ve Wi-Fi komunikaci a celkově síťové komunikaci je schopnost detekovat nové (zero-day) útoky.[1]

Datový bod je souhrn informací o nějakém předmětu, či události. Může se jednat např. o jeden řádek tabulky. Lze na něj pohlížet jako na bod v prostoru s počtem dimenzí rovným množství datových atributů (sloupců). Datový bod lze vyjádřit jakožto vektor.

Datové featury jsou parametry/proměnné/atributy dat.

Dimenzionalita dat vyjadřuje množství jednotlivých atribut.

Vysokodimenzionální data Data s vysokým počtem atributů.

Metody učení s učitelem Tyto metody vyžadují předem označená data, tj. v tomto případě, zda se jedná o anomální či normální data.[10]

Metody učení bez učitele Metody, které se spokojí s předem neoznačenými daty.[10]

2.1.2 Měření přenosti modelů

	Anomálie	Normální chování
Označeno za **anomální**	**TP**	**FP**
Označeno za **normální**	**FN**	**TN**

Obrázek 2.1: Možnosti výstupu klasifikace

Je důležité ujasnit dle jakých parametrů je definována přesnost modelu. Z principu detekce vyplývá, že model může detekovat normální chování správně, označujeme TN (True Negative), či špatně, označujeme FP (False Positive). Špatně klasifikované anomálie jsou označovány FN (False Negatives) a správné TP (True Positives).

FPR/FAR rate

$$FPR = \frac{FP}{FP + TN} \tag{2.1}$$

Počet chybně klasifikovaných dat anomálního chování ku celkovému počtu normálních dat. Také nazýváno Fall-out.

P_D (citlivost, TPR)

$$TPR = \frac{TP}{TP + FN} \tag{2.2}$$

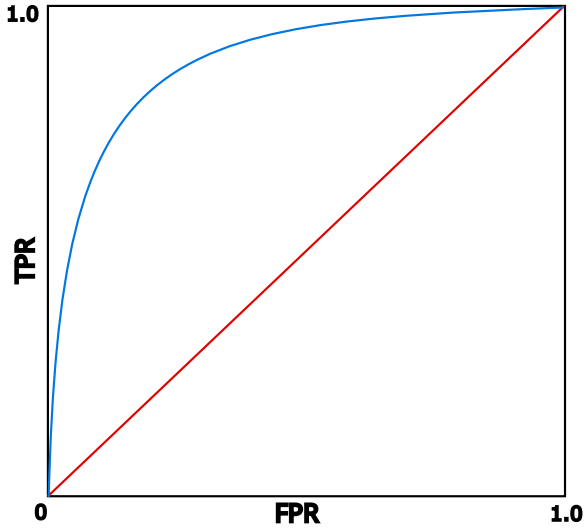
Množství správně klasifikovaných dat anomálního chování ku všem datům představujícím anomální chování.

ROC Receiver Operating Characteristic. Podle [22] se jedná se o křivku, která zobrazuje závislost FPR (osa x) a TPR (osa y), tj. $\mathbf{P_D}$. Často se stává, že je zapotřebí, aby FAR nepřesáhl určitou hranici, tudíž na základě požadovaného FAR je zvolen výsledný klasifikátor.[3] Ideální klasifikátor bude mít nulové FPR a 100% TPR.[22] Příklad ROC křivky lze vidět na obrázku 2.2.

F(1) Measurement

$$F(1) = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{2.3}$$

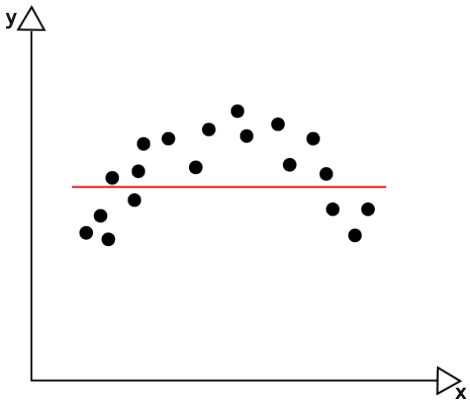
F(1) score je harmonický průměr TPR a PPV (Positive Predictive Value).[18]



Obrázek 2.2: Příklad ROC křivky. Ideální klasifikátor bude mít nulové FPR a 100% TPR.[22]

2.1.3 Bias a variance

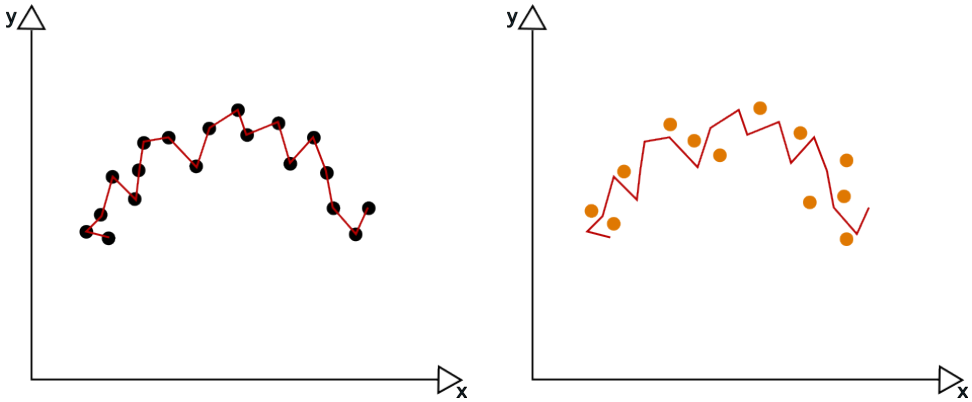
Bias a variance jsou dvě významné zdroje chyb. Bias vyjadřuje rozdíl mezi očekávánou odchylkou od skutečných hodnot. Bias se primárně projevuje chybou při učení na trénovacích sadě. Variance udává odchylku od očekávané hodnoty, způsobenou vzorkováním dat. Vysokou varianci lze poznat dle špatné schopnosti generalizace modelu na testovacím vzorku dat. V samé podstatě variance říká, jak dobře naučený model popisuje skutečnou distribuci dat.



Obrázek 2.3: Underfitting - vysoký bias.[10]

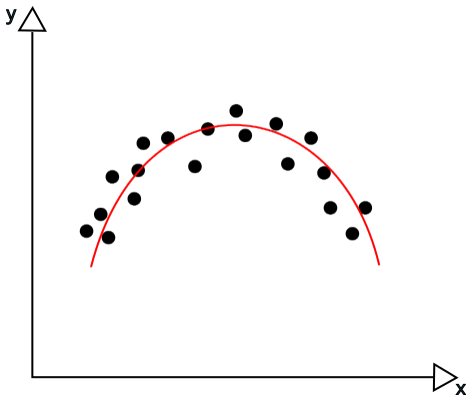
Underfitting Podle [10] vyjadřuje situaci, kdy model není schopen dosáhnout dostatečně nízké chybovosti při učení. V obrázku 2.3 lze vidět, jak se vysoký bias může projevit.

Overfitting nastává ve chvíli, kdy je příliš vysoký rozdíl mezi dosaženou, měřenou chybou při učení a chybou při testování.[10] Model jinými slovy špatně generalizuje. V obrázku 2.4



Obrázek 2.4: Overfitting - vysoká variace.[10] Na levém obrázku je vidět, že model perfektně modeluje každý datový bod. Na pravém obrázku je vidět model dat a datové body testovací sady. Model funguje výrazně hůř pro testovací sadu.

lze vidět napravo vysokou variaci při učení a vlevo jak se může tato vysoká variace projevit při testování.



Obrázek 2.5: Ideální rozpoložení variace a bias

Ideální bias a variace Při modelování se snažíme o nízký bias a variaci zároveň. V obrázku 2.5 lze vidět ideální (nízkou) variaci a bias modelu.[10]

Generalizace Vyjadřuje, jak dobře model funguje pro předem neviděná data.[10]
Snad nejdůležitější vlastností systému je jeho přesnost. Pokud systém až příliš často detekuje planné poplachy, přestávají upozornění zodpovědnou osobu zajímat. Systém v takovém případě přináší informaci o minimální hodnotě i při malé chybovosti, neboť je pouze malá šance, že klasifikovaná anomálie je skutečně anomálií (více: base-rate fallacy). Je však zároveň žádoucí, aby systém detekoval co možná nejvíce skutečných anomálií. Aby byl výsledný systém spolehlivý, je zapotřebí, aby dosahoval vysoké přenosti a velmi malé chybovosti ve skutečném provozu.[21]

2.2 Metody detekce anomálií založené na strojovém učení

2.2.1 Asociační pravidla - Association Rules

Datamining metoda, která hledá pravidla vyjadřující závislost mezi atributy. Používají metriky pro zjištění, jak často se vyskytuje určitý vztah mezi předměty v datech. Typické pravidlo vypadá způsobem *pokud je přítomno A a zároveň je přítomno B, je přítomno C.* Původně byla tato metoda používána pro hledání závislostí mezi předměty v nákupních datech.[3]

Odtud je odvozena hlavní nevýhoda asociačních pravidel, dokáže zpracovat pouze binární data, tedy zda se předmět v datech objevuje či ne. Tento nedostatek se snaží napravit její variace **fuzzy asociační pravidla**, která je schopna zpracovávat číselné a kategorické proměnné.[3]

Fuzzy asociační pravidla - Fuzzy Association Rules Modifikace klasických asociačních pravidel, která má možnost zpracovávat nebinární data. Pro účely detekce anomálií se používá převážně tato metoda. Typické pravidlo rozřazuje předmět podle jeho hodnoty. U binárních dat jsme se setkali s pravidlem, které začínalo *pokud je A přítomno.* Tato metoda však pracuje s nebinárními hodnotami, tudíž: *pokud A má tuto hodnotu/má hodnotu v tomto rozmezí.* Pravidlo by končilo způsobem *C má tuto hodnotu/má hodnotu v tomto rozmezí* místo *C je přítomno.*[3]

Výhody Pravidla lze jednoduše interpretovat. Je z nich očividné, jak detekce probíhá. Toto může být velmi užitečné pro analytiky, kteří hledají nové signatury útoků.[3]

Nevýhody Trpí na vysokým FP. Přestože jsou práce využívající této metody zmíněné v [3] hojně citované, jsou již více než 10 let staré a jejich výsledky se nedají srovnat s modernějšími přístupy.

2.2.2 Naivní Bayesovský klasifikátor - Naïve Bayes

Naivní Bayesovský klasifikátory spadají pod pravděpodobnostní klasifikátory aplikující Bayesovu větu. Naivním se nazývá, neboť předpokládá, že atributy jsou nezávislé. Podmíněné pravděpodobnosti vytváří klasifikační model, který přiřazuje třídu dle rovnice 2.4, převzáté z [3]:

$$y(f_1, f_2, \ldots, f_m) = argmax_{k\in\{1,\ldots K\}} p(C_k) \prod_{i=1}^{m} p(f_i|C_k) \tag{2.4}$$

Kde $\mathbf{m}$ je počet atributů, $\mathbf{K}$ je počet tříd, $\mathbf{f_i}$ je **i**-tá atribut, $\mathbf{C_k}$ je **i**-tá třída, $\mathbf{p(C_k)}$ je předchozí pravděpodobnost třídy $\mathbf{C_k}$ a $\mathbf{P(f_i|C_k)}$ je podmíněná pravděpodobnost atributu $\mathbf{f_i}$ v přítomnosti třídy $\mathbf{C_k}$.[3]

Naivní Bayesovský klasifikátor dokáže zpracovat libovolné množství nezávislých proměnných. Přestože má svá omezení, je ideálním klasifikátorem, pokud jsou atributy nezávislé. Většinou se spíše používá k porovnání s více sofistikovanými metodami.[3]

Výhody Naivní Bayesovský klasifikátor je výpočetně velmi jednoduchý na naučení a po trénování je schopen online nasazení.

Nevýhody Jednotlivé dimenze dat musí být nezávislé pro optimální fungování této metody. Nutnost této podmnínky podtrhují výsledky jedné z prací v [3], která vykazuje podprůměrnou přesnost a vysoký FAR.

2.2.3 Bayesovská síť - Bayesian Network

Bayesovská síť je pravděpodobnostní grafický model, který zobrazuje proměnné a vztahy mezi nimi. Síť se skládá z uzlů a hran, které reprezentují proměnné a jejich vztahy. Jedná se o orientovaný graf, kdy podřazené uzly jsou závislé na svých rodičích. Každý uzel uchovává stav náhodné proměnné a své podmíněné pravděpodobnosti. Bayesovské sítě lze konstruovat za pomocí expertních znalostí nebo za pomocí odvozovacích algoritmů, tj. učení se z dat. Hlavním úkolem Bayesovksých sítí je odvozování skrytých proměnných, hledání parametrů a vytváření struktury.[3][12]

Výhody Jsou schopny dosáhnout dobrých přesností. Mají výbornou rychlostí učení. Podle výsledků testů popsaných v [3] provedených na datovém setu KDD1999 dosahuje metoda vysokých přesností.

Nevýhody Vyžadují expertní znalosti k vytvoření základního modelu.

2.2.4 Shlukování - Clustering

Seskupování zahrnuje celou řadu technik, které jsou si vzájemně podobny zaměřením na vysokodimenzionální data. Jedná se o metody učení bez učitele, kde data jsou seskupovány podle míry podobnosti. Pro tyto metody není třeba poskytovat data rozdělená do předem definovaných tříd, tj. normální a anomálie.[3]

Existuje celá řada algoritmů spadajících do této kategorie. Následuje seznam různých podkategorií, do kterých lze algoritmy zařadit.[3]

- **Hiearchické modely** - data jsou spojovány do kategorií na základě vzájemné vzdálenosti
- **Centroidní modely** - řazení dat do tříd na základě podobnosti ku zvolenému bodu
- **Distribuční modely** - hledají vhodné statistické rozložení, které nejlépe odpovídá rozmístění datových bodů
- **Modely založené na hustotě** - shlukují body na základě lokální hustoty
- **Grafové modely** - po spojení bodů do grafové struktury jsou jejich závislosti analyzovány dle vhodného algoritmu

Výhody Metody mohou být snadné na implementaci a jsou schopny dosahovat dobrých výsledků.[3] zmiňuje práci z roku 2014, která na KDD datasetu dosáhla 98% přesnosti. Bohužel práce nezmiňuje přesný FAR.

Nevýhody Jedná pouze o metody učení bez učitele a obecně je těžké dosáhnout dobrých přesností.[3]

2.2.5 Evoluční výpočet - Evolutionary Computation

Evoluční výpočet popisuje kategorii mnoha genetických a optimalizačních algoritmů, z nichž jsou dvě nejpoužívanější Genetic Programming (GP) a Genetic Algorithms (GA). Skládají se z populace jednotlivců (chromozomů), fungující na bázi přežití nejschopnějšího. K určení nejschopnějších jedinců se používá fitness funkce, která vyjadřuje schopnost jedince řešit daný problém. Schopnější chromozomy mají větší šanci se množit. Chromozomy si mezi sebou vyměňují genetický materiál (postup řešení). Mutace je mechanismus, kterým se snaží algoritmus vyhnout lokálnímu minimu, jedná se o malou šanci náhodné změny v řešení. Mezi zmíněnými dvěma algoritmy je rozdíl v reprezentaci jedince. GA představuje jedince jakožto bitovou posloupnost, chromozomy GP zato obsahují operátory a programové bloky (if then, loop etc.).[3]

Výhody Vysokou potencionální přesnost metody GP dokazují výsledky jedné z prací citované v [3].

Nevýhody Největším problémem těchto modelů by se dala předpokládat složitost jejich přeučování, která je nejen výpočetně náročná, ale neexistuje žádná záruka, že novější řešení bude lepší. Tato problematika je podtrhnuta zveřejněním průměrných i nejlepších výsledků v [17], které se mohou výrazně lišit.[3]

2.2.6 Rozhodovací stromy - Decision Trees

Rozhodovací stromy se jako skutečné stromy skládají z větví a listů. Listy představují třídu, do které lze po větvích dojít. Větve představují spojení rozmezí hodnot atributů, které vedou k danému listu. Existuje řada metod k tvoření těchto stromů. ID_3 a $C_{4.5}$ jsou jedním z nich. Oba algoritmy využívají entropie, aby zvolily atribut, dle které se bude datový set na dalším větvením dělit. Pokud jsou takto vytvořené stromy malé, jsou velmi intuitivní a lze z nich snadno vyvodit pravidla. Velké stromy lze rozdělit na podstromy pro lepší přehlednost. Velké stromy také trpí špatnou schopností generalizace.[3] Algoritmy celkově jsou lehké na pochopení i implementaci.

Výhody Rozhodovací stromy jsou velmi lehké na implementaci. Nabízejí intuitivní pohled do mechanismu detekce. Dle výsledků z jedné ze studií v [3], lze touto metodou dosáhnout vysoké přenosti detekce.

Nevýhody U složitějších klasifikací, tj. větších vygenerovaných stromech, dochází ke špatné schopnosti generalizace. Algoritmy pro vytváření stromů také vždy inklinují k určitým atributům z důvodu využívání entropie pro jejich volbu.

Forest algoritmy

Jedná se o algoritmy, které kombinují rozhodovací stromy se skupinovým učením, tj. kombinují více rozhodovacích stromů. Jedním z představitelů této kategorie je Random Forest. Random Forest jako vstup přijímá náhodně vybrané atributy dat. Samotný Forest je tvořen jako kolekce rozhodovacích stromů s řízenou variací. Výsledek může být rozhodnut např. na základě většiny. S množstvím stromů se snižuje variance modelu, zatímco bias zůstává stejný. Random Forest je závislý na náhodném generátoru.[3]

Výhody Snížení variance oproti běžnému rozhodovacímu stromu. Modely jsou obecně odolnější vůči over-fittingu. Není třeba volit atributy dat dle důležitosti, neboť je model schopný zpracovat vysoké množství parametrů sám.[3]

Nevýhody Interpretovatelnost výsledků je obtížná.[3] Dochází oproti původnímu rozhodovacímu stromu ke ztrátě výkonnosti.

2.2.7 Metoda podpůrných vektorů - Support Vector Machine

Support Vector Machine (SVM) hledá nadrovinu, kterou by rozdělil prostor proměnných tak, aby byla vzdálenost k nejbližším datovému bodu co největší. Tento přístup je založen na minimálním klasifikačním riziku, ne na optimální klasifikaci. Nejlépe fungují pro datové sady s nízkým počtem dat, velké množství dat je nežádoucí, neboť značně prodlužuje dobu tréninku. Dokážou se také snadno vyrovnat s vysokodimenzionálními daty. SVM mají skvělou schopnost generalizace, trpí však, pokud je značný nepoměr v množství dat různých tříd, tento problém se dá vyřešit váhováním tříd.

Výhody SVM má skvělou schopnost generalizace a je schopna efektivně pracovat s vysokodimenzionálními daty.

Nevýhody Hlavní nevýhodou SVM je doba tréninku, která se může s množstvím dat výrazně prodloužit.

2.2.8 Skupinové učení - Ensemble Learning

Jedná se techniku aplikující princip "dvě hlavy jsou lepší než jedna". Spočívá v použití více metod pro klasifikaci. Obecně využívá více slabších metod pro vytvoření silnějšího modelu [3], není to však pravidlem. Jedním z nejznámějších modelů, aplikujících tuto techniku je Random Forest.

2.2.9 Umělé Neuronové sítě - Artificial Neural Networks ANN

Technologie založená na skutečném způsobu fungování neuronových sítí (dále NN). Způsob implementace se nejlépe popisuje na perceptronu, což je předek novodobých neuronových sítí (dříve byli nazývané jako vícevrstvé perceptrony).

Perceptron má určitý počet vstupů a jeden výstup. S každým z těchto vstupů je spojena určitá váha. Samotný perceptron je v samé podstatě aktivační funkce, která určí velikost výstupu na základě součtu vstupů vynásobených jejich vahami. Funkce používaná pro aktivaci perceptronu se nazývá jednotkový skok. Pokud propojíme více perceptronů po vrstvách, vzniká neuronová síť.

Klíčová schopnost NN pro její funkčnost je back-propagation. NN za pomocí hodnotící funkce zpětně upraví váhy vstupů jednotlivých neuronů na základě gradientu. Hodnotící funkce určuje, jak blízko byl výsledek výsledku ideálnímu.[3]

V poslední době znovu vzrostly na popularitě díky vývoji nových metod užívání a také častým výhrám v soutěžích rozpoznávání vzorů.[3]

Výhody Jedná se o velmi přesnou metodu pro aproximaci libovolné funkce.[10]

Nevýhody S komplexitou neuronové sítě se výrazně zvyšují nároky na výpočetní výkon, obvzláště u modernějších, složitějších verzí neuronových sítí.[3] S dobou tréninku lze bojovat pomocí trénování modelu na grafické kartě.
NN mohou trpět na lokální minima, neboť ideální funkci hledají pomocí gradientního sestupu. Tento problém lze zmírnit pomocí vhodné volby parametrů NN.

Varianta NN, LSTM rekurzivní neuronová síť, byla použita pro implementaci klasifikátoru systému v této práci. LSTM rekurzivní neuronová síťe jsou dále popsány v podkapitole 2.3.

2.3 LSTM rekurzivní neuronové sítě

Jedná se o variantu běžné neuronové sítě, popsané v předchozí kapitole, v sekci 2.2.9. V této podkapitole bude stručně popsána teorie LSTM RNN (Long Short-Term Memory Recursive Neural Network). Metoda je použita jako implementace modelu systému navrženého v této práci. Přesnou teorii s matematickým základem lze nalézt v [20].

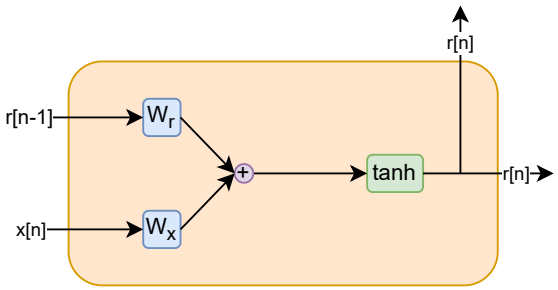
2.3.1 Rekurzivní neuronové sítě

Rekurzivní neuronové sítě vznikly pro modelování sekvečních dat, tedy dat, u kterých je hodnota datového bodu závislá na předchozích bodech.

$$\overrightarrow{s}[n] = W_r \overrightarrow{r}[n-1] + W_x \overrightarrow{x}[n] + \overrightarrow{\theta_s} \tag{2.5}$$

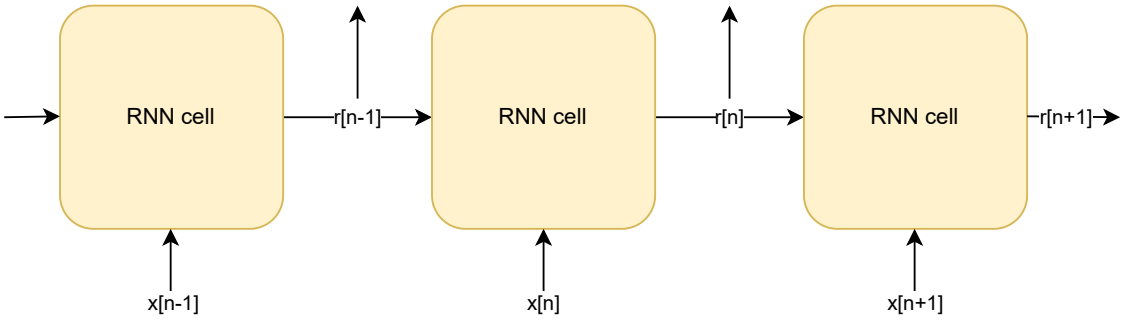
$$\overrightarrow{r}[n] = G(\overrightarrow{s}[n]) \tag{2.6}$$

Rovnice 2.5 a 2.6 z [20] popisují rekurzivní neuronovou síť, kde $\overrightarrow{s}[n]$ značí stav, $\overrightarrow{r}[n-1]$ výstup předchozí buňky v sekvenci a $\overrightarrow{x}[n]$ vstup aktuální buňky. $\overrightarrow{r}[n]$ je výstup aktuální buňky, G(x) je aktivační funkce, $\overrightarrow{\theta}$ představuje tzv. bias a W_r a W_x jsou matice vah. 2.6 vyjadřuje celou rekurzivní neuronovou síť. Ve 2.6 lze vidět vizuální popis RNN buňky dle



Obrázek 2.6: Popis RNN buňky pomocí diagramu dle specifikace v [20]. θ byla vynechána pro přehlednost.

[20]. RNN neuronovou síť můžeme tzv. rozbalit, jak lze vidět v 2.7. Celkově síť dokáže udržet kontext pouze K kroků, kde K je velikost sekvence vstupů. Pro zachycení dlouhodobých závislostí je potřeba zvětšit velikost okna. Toto však může způsobit tzv. vanishing a exploding gradient. [20]

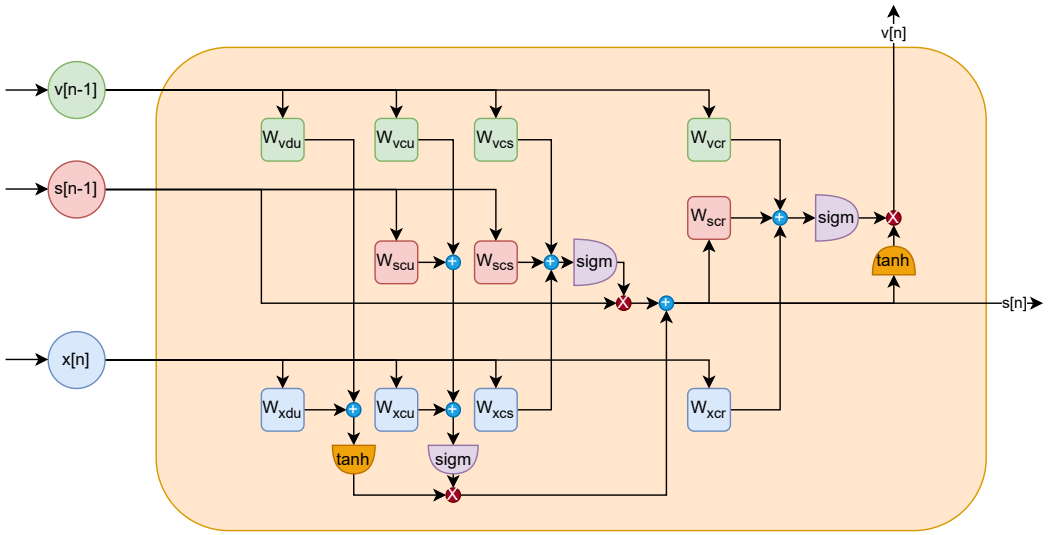


Obrázek 2.7: Vizualizace rozvinutí RNN podle rovnic 2.5 a 2.6

2.3.2 Long Short-Term Memory Rekurzivní Neuronové Sítě

Long Short-Term Memory RNN (LSTM) bojují proti vanishing a exploding gradientu. [20] poskytuje detailní popis matematického základu pro LSTM. V této práci bude LSTM popsáno pouze pomocí grafického znázornění 2.8, protože matematický popis je zdlouhavý a složitý. Nejpodstatnější části LSTM jsou hradla, které upravují stav buňky. Základní LSTM používá 3 typy těchto hradel:

- Řízení množství informací převzatých z předchozího stavu.
- Řízení množství informací pro aktualizaci aktuálního stavu, odvozeného z předchozího stavu, aktuálního vstupu a výstupu předchozí buňky.
- Řízení výstupu buňky, který je odvozen z aktuálního stavu buňky, aktuálního vstupu a výstupu předchozí buňky. Tato hodnota je přístupná další buňce v sekvenci.



Obrázek 2.8: Vizualizace klasické LSTM podle grafického vyobrazení z [20]

2.8 popisuje LSTM buňku pomocí diagramu. Vstupem buňky jsou featury aktuální části sekvence *x[n]*, výstup předchozí buňky *v[n-1]* a stav předchozí buňky *s[n-1]*. Výstupem jsou textits[n], který znační stav aktuální buňky, a *v[n]*, který je výstupní hodnotou buňky a

který je posílán do další vrstvy buněk a zároveň další buňce v sekvenci. Přesné významy jednotlivých komponent lze vyčíst z [20]. LSTM je netriviální metoda, která je však spolehlivější pro dlouhodobé závislosti v datech.[20]

Kapitola 3

IEEE 802.11

IEEE 802.11 [13] spadá do kategorie bezdrátových LAN. Implementace splňující podmínky tohoto standardu běžně nazýváme Wi-Fi (Wireless-Fidelity). Wi-Fi jsou jednou z nejpopulárnějších možností připojení k Internetu pro běžného uživatele. Různé verze tohoto standardu se snaží doplnit neduhy verzí předchozích nebo vyplnit mezeru v poptávce. V základu se však jedná o bezdrátovou technologii podobnou Ethernetu. Podobnou především proto, že si mnoho svých technologií od Ethernetu vypůjčuje.[16]

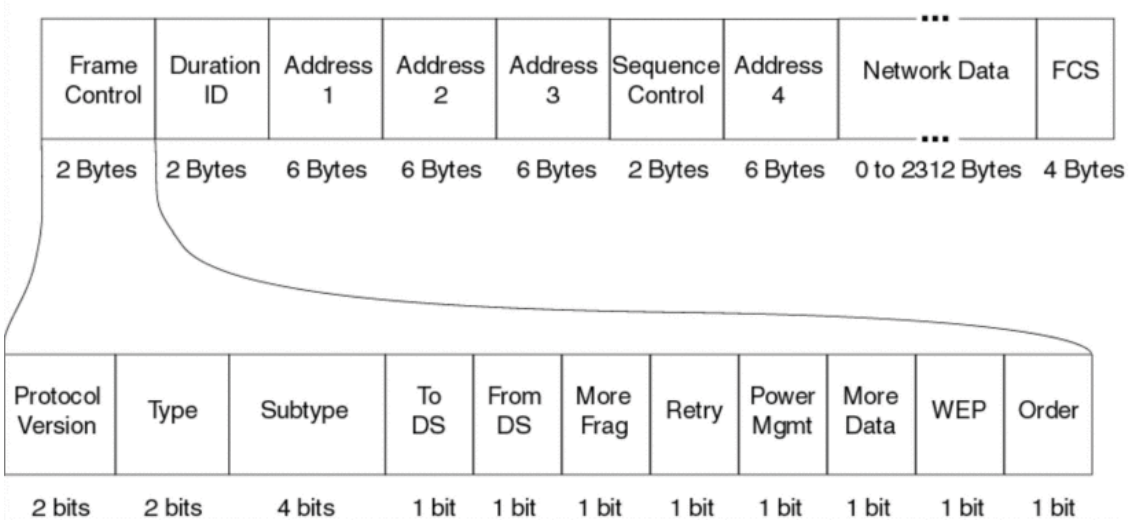
3.1 Rozdíl mezi Ethernet a IEEE 802.11

Přestože si IEEE 802.11 od Ethernet půjčuje mnoho principů a technik, nejsou totožné. Hlavním rozdílem je, že Wi-fi je bezdrátovou technologií, což se projevuje především v typické infrastruktuře sítě Wi-Fi, kde se koncové stanice připojují k Access Pointu (AP). Kolekce stanic spolu s jedním AP se nazývá Basic Service Set (BSS). BSS je většinou přes své AP připojen k routeru, který celou podsíť propojuje se zbytkem sítě. BSS lze kombinovat v rámci jedné sítě do Extended Service Setu (ESS).[16]

Na úrovni samotných rámců se technologie výrazně liší převážně v hlavičkách. Hlavička IEEE 802.11 obsahuje 4 políčka pro adresu, na rozdíl od Ethernet, který obsahuje 2, a tzv. frame control, který poskytuje velkou většinu funkčnosti BSS pro zprávu komunikace. 3 políčka jsou nutné pro běžnou komunikaci, neboť AP je zařízení na linkovací úrovni, tudíž nedokáže routovat provoz. 4. adresa slouží pro ad hoc sítě.[16] Hlavičku IEEE 802.11 lze vidět v obrázku 3.1 převzatého z [19].

Největší a nejzávažnější rozdíl mezi Ethernet a IEEE 802.11, je v užitém médiu pro komunikaci. Zatímco do sítě Ethernet je zapotřebí se fyzicky připojit ke přepínači, což samo o sobě poskytuje určitou formu zabezpečen, do sítě Wi-Fi se může připojit kdokoliv v dostatečné vzdálenosti od AP. Stejně jak je možno se připojit, tak je možno naslouchat. V síti LAN má člověk možnost naslouchat komunikaci až po připojení k přepínači, naopak v síti Wi-Fi může naslouchat komunikaci kdokoliv, kdo je v efektivním dosahu vysílání. Toto představuje bezpečnostní riziko pro všechny standardy IEEE 802.11.[21]

Technologie se dále liší v použitém protokolu pro správu přístupu k médiu. Ethernet používá CSMA/CD (Collision Detection), vzato IEEE 802.11 užívá CSMA/CA (Collision Avoidance).



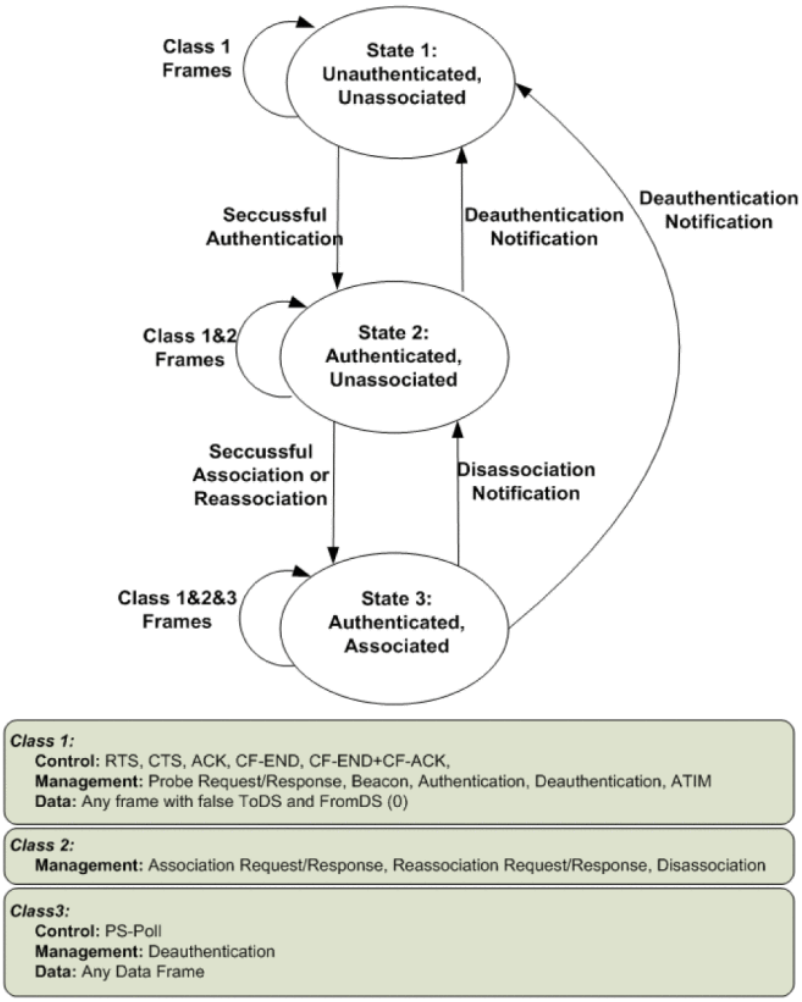
Obrázek 3.1: Hlavička 802.11 ze [19]

CSMA/CA funguje na principu vyvarování se kolize. Zjednodušeně, pokud stanice zjistí, že kanál, na kterém chce vysílat, není volný, počká s jakýmkoliv vysíláním náhodně vybranou dobu, po které zkusí vysílat znova. S tímto jsou spojeny další techniky 802.11, RTS (Request To Send), kterým stanice žádá o povolení zaslat větší množství dat, a CTS (Clear To Send) tuto akci povoluje. RTS se používá, aby stanice ohlásila, že bude kanál používat delší dobu, ostatní stanice by se měly na oplátku vyvarovat tento dlouhý řetězec dat poškodit svým vlastním vysíláním. Lze tohoto využít k formě DOS útoku.[16]

Připojení k síti Wi-Fi se provádí skenováním jednotlivých kanálů. O připojení se lze hlásit aktivně, vysláním broadcast packetu po kanálech a počkáním na odpověď, nebo pasivně, kdy každé AP typicky vysílá svůj vlastní broadcast packet každých 100 ms. Připojování k Wi-Fi je principiálně podobné připojování k DHCP serveru.[16] Stavový automat 802.11 komunikace lze vidět v 3.2. Obrázek je převzat z [1].

3.2 Bezpečnostní standardy Wi-Fi

Protože jsou Wi-Fi zařízení principielně náchylnější jako cíle útoků, existuje řada standardů, které pomáhají sítě a klienty chránit. Prvním z těchto standardů - WEP (Wired Equivalent Privacy) - byl představen v původním IEEE 802.11 a byl vyřazen v roce 2004, přesto je však stále používán malým počtem sítí [8] a útoky na něj jsou stále relevantní. V roce 2003 přišel na svět WPA (Wi-Fi Protected Access) jako předzvěst WPA2, který byl představen následující rok. Jako odpověď na řadu útoků, kterým podléhal WPA2, v roce 2018 Wi-Fi Alliance představilo WPA3 certifikaci.
Na obrázku 3.3 lze vidět procentuální zastoupení bezpečnostních protokolů napříč sítěmi v Budapešti z roku 2018. Značná část sítí nepoužívá žádné zabezpečení a můžou se tak stát snadnými cíly útoků.



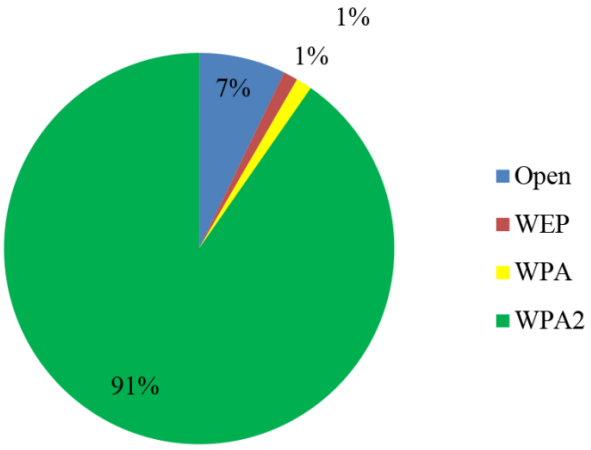
Obrázek 3.2: Stavový automat 802.11 z práce [1]

3.2.1 WEP

WEP používá stream šifrování RC_4 a CRC-32 kontrolní součet. Pro RC_4 klíč byl použit mimo jiné 24bitový vstupní vektor, který je generován pro každý paket a sloučen s tzv. root key. Root key je statický klíč, který lze nastavit v AP. Autentikace WEP může být Open System (neprobíhá autentikace) nebo Shared Key. Autentikace pomocí sdíleného klíče probíhá dle 4krokového challenge-response handshake. Vstupní vektor a RC_4 šifrovaní se stalo cílem řady útoků. WEP také nepodporuje autentikaci AP vůči klientovi.[15]

3.2.2 WPA a WPA2

WPA pro šifrování využívá TKIP (Temporal Key Integrity Protocol) nebo AES jako alternativu. Snaží se zmírnit bezpečnostní dopady WEP použitím vhodnější délky vstupních vektorů a silnější kontroly integrity (mechanismus Michael). WPA je založena na práci s RADIUS serverem. Alternativou pro domácnosti je WPA-PSK (WPA Pre-Shared Key), které je zjednodušenou variantou WPA a je funkčně velmi podobné WEP.[15]
WPA2 zavádí hiearchii klíčů. Na vrcholu hiearchie existuje buď PSK (Pre-Shared Key) nebo MSK (Master Session Key). O WPA2 se lze více dozvědět ve [14]. Důležité je zmínit,



Obrázek 3.3: Procentuální zastoupení bezpečnostních protokolů v Budapešti roku 2018 z práce [8]

že řada útoků na WPA2 využívá zpětné kompatibility s WPA. KRACK útočí na 4-way handshake WPA2.[23]
Největší slabinou WPA2 je nezabezpečení management rámců a neschopnost bránit se útokům na dostupnost. 802.11w se věnuje těmto problémům a představilo RMF (Robust Management Frames), které zabezpečují management rámce, a řadu dalších mechanismů pro boj s asociačními útoky.[15] Z obrázku 3.3 lze vidět, že značná část sítí v tomto roce používala WPA2 zabezpečení. Část sítí, které používaly WPA2 zabezpečení, ale bez podpory 802.11w standardu, se mohly stát snadnými cíly útoků, přestože byly zabezpečeny.

3.2.3 WPA3

WPA3 vzniklo jakožto odevza na KRACK útok WPA2. WPA3 zavádí řadu protokolů, které musí zařízení podporovat. Obecně WPA3 zavádí Dragonfly handshake a zpětnou kompatibilitu WPA3 s WPA2. Dragonfly handshake má bránit proti offline slovníkovým útokům. Dragonfly handshake varianta ve WPA3 je také známá jako SAE (Simultaneous Authentication of Equals).[24]

3.3 Útoky IEEE 802.11

Tato sekce se bude zabývat různými útoky na IEEE 802.11 a jeho bezpečnostní standardy (WEP, WPA, WPA2). Jelikož se tato práce zabývá pouze OSI vrstvou L2 (data link layer), nebudou brány v potaz útoky na nižší a vyšší vrstvy. Dále je brán důraz na útoky, které jsou praktické a snadno aplikovatelné, čili útoky, které lze snadno a efektivně provést. Jedná se zejména o útoky pro jejichž použití existují již implementované nástroje.
Rozdělení a jednotlivé útoky jsou výtahem z [15] a shrnutí útoků podle autora stejné práce lze najít v příloze B.2.

3.3.1 Útoky na klíče bezpečnostních mechanismů IEEE 802.11

Útoky na tajný klíč komunikace se snaží nabourat důvěrnost komunikace. Tyto útoky lze provést pasivním způsobem, kdy ke lámání klíče dochází offline. Takové útoky je v podstatě nemožné detekovat. Podle [15] však útočník často zvolí aktivní způsob útoky, při kterém dochází k zaslání velkého počtu rámců určitého typu.

FMS a KoreK útoky

Korek a FMS útoky jsou útoky na WEP klíče. Přestože je WEP zastaralý, stále je používán v řadě zařízení.[15] FMS útočí na RC_4 Key Scheduling Algorithm (KSA). Konkrétněji se pokouší odhadnout n+1 byte šifrovacího klíče. Detailní popis útoku lze nalézt v [9]. KoreK rodina útoků používá matematické principy [9] a statistické metody pro zvýšení pravděpodobnosti určení WEP klíče.[15]
Oba útoky mohou být užity zároveň. V tomto případě často náseduje brute force útok pro nalezení klíče. Tento přístup je používán pro zvýšení časové efektivnosti útoku.[15]

ARP injection útoky

Ve skutečnosti se nejedná o útoky, ale často jim předchází. Tento útok se pokouší donutit síť, aby generovala nové input vektory, které následně vkládá do algoritmu pro lámání WEP klíčů. Takto použité ARP Request pakety mívají broadcast adresu, avšak pokud útočník zná topologii sítě, tak může použít adresu skutečné stanice náležící této síti. Výhoda druhého přístupu spočívá ve vyhodnocení ARP Request, dohromady tak můžou vzniknout 3 vstupní vektory.[15]

PTW útok je efektivnější než dříve zmíněné statistické metody. V praxi využívá ARP injection a je často vychozí metodou řady nástrojů pro lámání WEP klíčů.[15]

Slovníkové útoky

Slovníkové útoky jsou používány pro lámání WPA a WPAP2 klíčů. Přestože je lze použít pro lámání WEP klíčů, tak nejsou používány, protože existují efektivnější metody.
Slovníkový útok je brute force útok, který používá databázi možných hesel, které porovnává s odposlechnutým 4-way handshake. Tento 4-way handshake může odposlouchávat pasivně nebo k němu stanici donutit pomocí zanedbatelného množství fabrikovaných Deauthentication rámců, kterými stanici od sítě odpojí. Útok je závislý na úplnosti databáze hesel a výpočetním výkonu útočníka. Jeho kritická část probíhá offline, což jej dělá takřka nedetekovatelným, i když násilím odpojí stanici od sítě. Lze jej však použít pouze na sítě zabezpečené pomocí PSK.

3.3.2 Keystream útoky

Alternativou pro útoky na sdílené klíče jsou útoky na keystream, kterým jsou šifrovány pakety. Útočník může keystream použít, aby fabrikoval pakety, jakožto odrazový můstek k dalším útokům. Může také dešifrovat části paketů, aby se naučil topologii sítě.[15]

Chopchop a fragmentační útoky

Chopchop využívá chybného využití CRC-32 a skutečnosti, že WEP nenabízí ochranu proti replay útokům. Jméno útoku je odvozeno od "odsekávání" posledního bytu šifrované části paketu. Útočník spoléhá, že bude informován AP o neplatnosti paketu. Podle [15] útočník AP používá jako vědmu. Útočník postupně zkouší různé XOR zkraceného paketu a průměru potřebuje 128*m pokusů pro odhadnutí posledních m bytů šifrované části paketu. Často se tento útok využívá pro odvození velké části keystream, tudíž vzniká zvýšený počet neplatných ICV rámců v komunikaci. Fragmentační útok se snaží být efektivnější a zaslat méně paketů než Chopchop útok. Pro své účely využívá možnost fragmentace poskytnutou v rámci IEEE 802.11 a LLC hlavičky, jejíž prvních 8 bytů je velmi snadno odhadnutelných.[15] Předpokládá se, že útočník se již úspešně autentizoval do sítě. Dále pomocí alespoň jednoho data paketu zkonstruuje pakety, které označí jako fragmenty, které si nechá po rekonstrukci zaslat zpět pomocí AP skrz broadcast adresu. Jelikož obsah paketů útočník předem zná, stačí mu provést XOR nad původními daty a rekonstruovaným paketem.[15] Při tomto útoku tedy dochází ke zvýšení počtu fragmentovaných paketů v síti.

Caffe Latte útok

Podle [15] je Caffe Latte útok směřovaný na WEP klíče. Využívá skutečnosti, že řada zařízení aktivně zasílá Probe Requesty hledající sítě, ke kterým se již v minulosti připojila. Jelikož WEP nenabízí možnost autentikace AP vůči stanici, nic nebrání útočníkovi vytvořit falešné AP s ESSID hledaného AP. Útočník potom počká než si stanice přiřadí vlastní IP v absenci DHCP serveru a začne zasílat ARP pakety. Tyto ARP pakety začne útočník modifikovat na ARP Requesty, kterými se snaží uhádnout IP adresu stanice. Jakmile dostane od stanice odpoveď, což indikuje úspěšné uhádnutí IP adresy, začne útočník posílat další ARP Requesty směřované na danou stanici. Takto útočník získá další vstupní vektory, které může použít pro metody lámání WEP. Jelikož tento probíhá mimo cílenou síť, je takřka nemožné jej detekovat.

Hirte útok je útok podobný Caffe Latte, který používá metod užitých ve fragmentačních útocích. Získaný ARP paket fragmentuje a přehází pořadí. Při tomto útoku dochází k náplavě jak ARP, tak fragmentovaných paketů.

KRACK útok

KRACK útočí na WPA2 4-way handshake a zpětnou kompatibilitu s WPA. Manipulací a replay útokem donutí znovupoužití předchozího klíče pro generaci keystream. Tento útok používá řadu různých zranitelností WPA2. Od použitých zranitelností se také odvíjí složitost jeho detekce, ale obecně se jedná o velmi zákeřný útok.[8] Tento útok funguje proti všem moderním Wi-Fi, protože se jedná o chybu přímo ve WPA2 standardu. Útok samotný se projevuje zvýšeným počtem zpráv 3. části 4-way handshake.[23]

3.3.3 Útoky na dostupnost sítě

Útoky na dostupnost, obecně DoS (Denial of Service), zabraňují v používání sítě jednotlivým stanicím nebo všem stanicím v určité oblasti. Podle [15] se jedná převážně o útoky využívající nechráněné management rámce. Aby však mohl být DoS útok proveden, musí

být útočník fyzicky v oblasti sítě. DoS útoky na IEEE 802.11 obecně také nemají dlouhodobé následky.

Deautentikační útoky

Deautentikační útok je jednoduchý a velmi efektivní útok na 802.11 sítě. Jakmile stanice přijme deautentikační rámec, musí okamžitě opustit síť.[15] Útočník jednoduše odposlechne MAC adresu stanice a deautentikační rámec zašle přímo stanici nebo AP, které stanici vypoví přístup k službám sítě. Stanice se může jednoduše znovu připojit k AP, ale nic nebrání útočníkovi zaslat více deautentikačních rámců.[15] Tento útok může být místo DoS použit jako odrazový můstek pro odposlechnutí připojení dané stanice do sítě.
Alternativně může útočník jako cílovou adresu zadat broadcast. Tímto způsobem se odpojí všechny stanice od sítě a následně se znovu pokusí připojit.[15]

Disasociační útok je útok provedením totožný s deautentikačními útoky. Jediný rozdíl je, že útočník zasílá disasociační rámce. Tento útok je teoreticky méně efektivní než deautentikační útok, protože je pro stanici snadnější se znovu připojit do sítě. Tento útok má taktéž broadcast variantu.[15]

Block ACK Flood

Block ACK Flood je velmi zákeřný útok na sítě podporující 802.11n. Útočník fabrikuje ADDBA rámec s velmi vysokým sekvenčním číslem, který pošle AP namísto cíle útoku. AP bude zahazovat všechny pakety, dokud rámce zaslané cílovou stanicí nedosáhnou fabrikovaného sekvenčního čísla. Tento útok je jen těžko detekovatelný, protože pro svou účinnost potřebuje jen minimální počet rámců. Ke všemu útočník ani nepotřebuje být přítomen po celou dobu průběhu útoku.[15]

Authentication Request Flooding útok

Útok využívá paměťové omezení na maximální počet položek v tabulce asociace klientů. Tato tabulka může být omezena pevně zakódovanou hodnotou nebo přímo hardwarovou kapacitou paměti. Útočník fabrikuje velké množství authentication request rámců s různými mac adresami stanic. Jakmile je tabulka zaplněná, AP může ztratit schopnost asociovat skutečné stanice. Je nutné podotknout, že stanice nemusí dokončit autentikační proces, aby byla do tabulky vložena. Tento útok se projevuje zvýšeným počtem autentikačních rámců.[15]

Falešný Power Saving útok

Tento útok využívá funkcionalitu power saving režimu 802.11. Útočník na místo stanice zašle AP null rámec oznamující vstup do power saving režimu. Dané AP začne schraňovat rámce cílené na stanici do bufferu. Následující beacon rámce poté budou obsahovat MAC adresu stanice v TIM části hlavičky, kterou bude stanice ignorovat, protože ve skutečnosti není v power saving režimu. Po nějaké době AP zahodí všechny data uložená v bufferu pro danou stanici. Tento útok se taktéž zasílá malé množství rámců pro splnění účelu, a tak je velmi obtížně detekován.[15]

CTS Flooding útok

Je útok na sítě, které používají CTS a RTS rámce pro kontrolu přístupu k sdílenému médiu. Útočník zasílá CTS rámce sám sobě nebo jiné stanici na místo AP a tak nutí ostatní stanice, aby odkládaly svůj přenos. Jedná se o principielně jednoduchý útok, který zaplaví síť CTS rámci.

RTS varianta Útočník místo CTS rámců zasílá velké množství RTS rámců, které žádají o dlouhé přenosové okno. Útočník se snaží vyštvat ostatní stanice z komunikace. Tato verze se projevuje zvýšeným počtem RTS rámců.[15]

Beacon Flooding útok

Beacon flooding používá beacon rámce pro DoS dvěma způsoby. První verzi je útok použit pro zaplnění seznamu dostupných ESSID (dostupných sítí) náhodnými ESSID. Druhá verze využívá specifické ESSID s různými neexistujícími BSSID. Druhá verze útoku spoléhá na implementaci stanice, která v ideálním případě začne kontrolovat, které ESSID patří ke skutečné síti.[15] Obě verze útoku se projeví zvýšeným počtem beacon rámců.

Probe Request Flooding útok

Tento útok používá nadměrné množství probe request rámců, aby zahltil AP, které je podle IEEE 802.11 povinné odpovědět na každý z nich.[15]

Probe Response varianta používá probe response rámce cílené na určitou stanici, která předem zaslala probe request. Útočník svému cíli poskytne nesmyslné informace o síti a tak mu zamezí přístup k ní.[15]

Man-in-the-Middle útoky

Man-in-the-Middle útoky se u 802.11 projevují převážně nastraženými AP. Útočník se snaží, aby se klient nevědomky připojil ke škodlivému AP. Útočník následně odposlouchává komunikaci klienta nebo může využít nastraženého AP, aby donutil klienta vyzradit klíčové informace o jiné síti.

Honeypot

Honeypot není útok sám o sobe. Útočník vytvoří síť, do které láká nové uživatele. O těchto uživatelích poté může sbírat informace, jelikož může vidět nešifrovanou komunikaci. Pokud útoky probíhají, tak probíhají na vyšších úrovních než OSI L2. Tento útok je jen těžko detekovatelný, spíše je na uživateli, aby se nepřipojoval do podezřelých sítí.[15]

Evil Twin

Evil Twin napodobuje skutečné ESSID. Jelikož může být více AP ve stejné oblasti se stejným ESSID a stanice se typicky připojuje k tomu, které má nejsilnější signál, je jednoduché pro útočníka nastavit vlastní AP s ESSID sítě. Pokud síť má heslo, měl by ho útočník nejdříve obdržet, aby nevyvolal podezření na straně oběti útoku. Jakmile je stanice připojena, může útočník komunikaci oběti monitorovat nebo může spustit útoky na vyšších úrovních OSI.[15] Tento útok jde teoreticky detekovat na základě vzniku nového AP.

Rogue Access Point

Rogue Access Points jsou neautorizované AP. Mohou být instalované jako zadní vrátka do sítě, např. zaměstnancem, bez vědomí administrátora. Typicky je AP připojené kabelem do sítě. Tento útok je taktéž těžko detekovatelný.[15]

3.3.4 Útoky na IEEE 802.11 WPA3

Autor KRACK zveřenil práci Dragonblood [24], ve které se zabývá útoky na WPA3 a analýzou Dragonfly handshake. V této práci mimo jiné představili řadu útoků na Wi-Fi komunikaci samotnou.

Slovníkový útok na WPA3 Útočník propaguje síť jakožto WPA2 síť místo WPA3. Přestože WPA3 je schopno detekovat downgrade útoky, protože v jedné ze zpráv dragonfly handshake je také seznam používaných šifer, útočníkovi stačí pouze jeden 4-way handshake, aby mohl začít se slovníkovým útokem.

DoS útok na WPA3 [24] přišel se jednoduchým mechanismem, jak obejít DoS ochranu WPA3. Útočník se vydává za klienta a vysílá commit rámce a reflektuje všechny tajné cookies, které k němu přijdou. Tento útok je schopen zahltit CPU AP.

Kapitola 4

Analýza požadavků na systém a návrh systému

Tato kapitola se zabývá analýzou požadavků na cílový systém, výběrem metody pro implementaci detekce anomálií a návrhem cílového systému. V první části je shrnutí požádavků na systém, druhá část se bude zabývat návrhem systému.

4.1 Analýza požádavků na systém

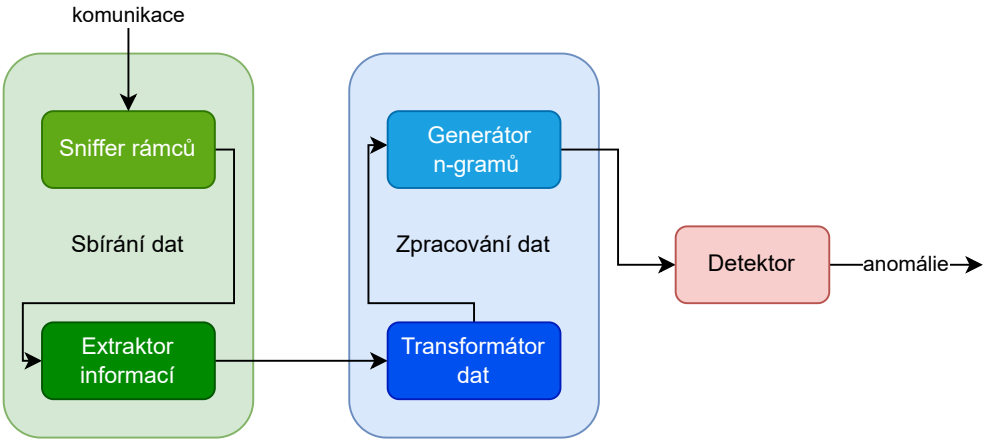
Nejpodstatnější vlastností navrhovaného systému je schopnost detekovat útoky a podezřelou komunikaci. V předchozí kapitole o 802.11 jsou shrnuty nejvýznamější útoky, které by měly sloužit jako průvodce návrhem systému. Pokud bude model schopen spolehlivě detekovat zejména DoS útoky, bude již detekovat více než polovinu běžných útoků na Wi-Fi zařízení a klienty. Velká část útoků, které nespadají do kategorie DoS se také projevují zvýšeným počtem rámců v komunikaci stejně jako DoS útoky.
Další podstatnou vlastností je schopnost detektoru se učit na neoznačených datech. Trénovací i testovací sada se bude skládat z komunikace realného provozu.

4.2 Návrh systému pro detekci anomálií

Návrh systému byl inspirován prací [19]. Celý systém se skládá z komponent pro sbírání dat, zpracování dat a detekci. V této podkapitole je popsán návrh systému po jednolitých částech, jak je lze vidět v diagramu 4.1.

4.2.1 Užité nástroje, jazyky a knihovny

Pro implementaci plánuji použít *Python* jako zvolený jazyk. Pro manipulaci s daty bude použita knihovna *Pandas* a *NumPy*. Pro implementaci neuronové sítě bude použita nadstavba knihovny *Keras TensorFlow*. Pro zachycení, export a analýzu dat bude použit nástroj *Wireshark*.
Pro vizualizaci dat bude použita knihovna *Matplotlib*. Pro vyhodnocení modelu knihovna *Scikit-Learn*. Pro uložení nastavení systému bude použita knihovna *PyYAML*.



Obrázek 4.1: Diagram systému pro detekci anomálií.

4.2.2 Sniffer rámců

Sniffer rámců má za úkol sbírat a ukládat rámce zachycené síťovou kartou. Pro tuto komponentu bude použit nástroj *Wireshark*. Existují alternativy jako *tshark*, *Kismet* a *airdump-ng*. *Wireshark* byl zvolen pro jeho schopnost vizualizovat rámce a snadnou extraci informací. Aby byl *Wireshark* schopen zachytit komunikaci, je zapotřebí nastavit libovolné bezdrátové rozhraní podporující Wi-Fi komunikaci do monitorovacího režimu a nastavit příslušný kanál, který je potřeba sledovat. Jak nastavit síťovou kartu do tohoto režimu je popsáno v podkapitole 5.1.
V monitorovacím režimu nemá rozhraní možnost vysílat, ale je schopné zachytávat libovolné rámce na daném kanálu.
V této práci budou použity data reálného provozu jak pro trénování, tak pro testování.

4.2.3 Extraktor informací

Rámce zachycené *Wireshark* mohou mít až 600 různých položek, podle [6] se však vyplatí vybrat pouze pár klíčových pro detekci. Pomocí nástroje *Wireshark* lze snadno extrahovat pouze důležité informace z rámců, jako je typ a podtyp rámce, source a destination adresy, který by bylo jinak nutné extrahovat ze 4 různých adres, a čas zachycení od počátku. Část možných typů rámců lze vidět v příloze B.1. Tabulka byla složena podle informací z [5].
Systém bude využívat pouze informace z L2 OSI vrstvy (linkové vrstvy). Tudíž pro detekci nejsou dostupné informace o síťové a dalších vrstvách, které jsou typicky šifrované.

4.2.4 Transformátor dat

Transformátor dat má za úkol předzpracovat data do vhodné podoby, aby je bylo možné vyhodnotit pomocí neuronové sítě. V základu se jedná o transformaci dat na číselnou hodnotu. V případě navrhovaného systému bude především zapotřebí číselně vyhodnotit typ rámce a zaokrouhlit čas zachycení směrem dolů. Zaokrouhlení času zachycení směrem dolů rozdělí rámce do intervalů po 1 sekundě, takto bude snadné spočítat množství zachycených rámců v určitém intervalu. Typ rámce je typicky hexadecimální hodnota a tak se jedná o pouhou konverzi do decimálního tvaru. Takto rozdělené rámce lze použít pro tvorbu sekvencí typů v komunikaci.

Pro transformaci dat byly použity knihovny *Pandas* a *NumPy*, které obě dovolují efektivní a přehlednou práci s daty.

4.2.5 Detektor

Pro detekci bude potřeba metoda učení bez učitele. Vhodnou metodou pro sekvenční neoznačená data je LSTM RNN (Long Short-Term Memory Recursive Neural Network) tvarem podobnou autoenkodéru (nejedná se však přímo o autoenkóder). Jednou z podmínek užití tohoto modelu je poskytnutí sekvencí dat statické délky. Délka musí být dostatečná pro zachování kontextu, příliš dlouhé sekvence jsou však zbytečně výpočetně náročné. Velikost sekvence záleží na charakteru dat a musí být zvolena experimentálně pro daný model.
Během tréninku se model snaží rekonstruovat sekvenci na vstupu. Rekonstrukční chyba je velikost rozdílu mezi vstupem a výstupem neuronové sítě. Model se pomocí této rekonstrukční chyby učí. Při detekci se stejná chyba používá jako hodnocení anomálnosti jednotlivých sekvencí. Hodnocení anomálnosti bude možné vyobrazit do grafu.
Vzhledem k charakteru dat bude vhodné detekovat anomálie v komunikaci na základě její struktury a množství. Lze použít metodu skupinového učení a zkombinovat dvě neuronové sítě, kde každá bude přijímat sekvence jiného charakteru. Detektor takto dosáhne zvýšené čitelnosti, neboť bude jasné, který typ informací anomálii zachytil.
Pro implementaci modelu bude použita knihovna *TensorFlow*.
Pro trénování LSTM modelu bude použito prostředí *Google Colab*, které dovoluje trénovat modely neuronových sítí na grafických kartách. U LSTM, jejíž největší nevýhoda je vysoká doba trénování, je tato skutečnost vítaná. Výstup modelu bude vizualizován pomocí *Matplotlib*.

4.2.6 Generátor n-gramů

Pro detektor navržený v této práci bude zapotřebí vytvořit sekvence dat, tzv. n-gramy. N-gramy budou generovány pomocí klouzavého okna délky n ve formě základního *list* datového typu a *array* knihovny *NumPy*. Systém bude generovat n-gramy pro posloupnost jednotlivých rámců a počet typů zachycených rámců v časovém rozmezí.

Kapitola 5

Implementace systému pro detekci anomálií na základě návrhu

Tato kapitola bude popisovat implementaci systému dle návrhu. V podkapitole Sniffer je popsán způsob sbírání dat pro trénování a testování, nastavení síťové karty do monitorovacího režimu, samotný sběr dat, jak pro trénování, tak pro testování. V podkapitole Extraktor je popsána práce s nástrojem *Wireshark* a vyběr informací z rámců pro trénovací a testovací data. V další podkapitole je stručně popsána práce s Google Colab s grafickým doprovodem. V podkapitole o předzpracování dat jsou vysvětleny jednotlivé způsoby zpracování dat a funkce a třídy k tomu užité. V následující části je popsána implementace modelu LSTM RNN a následně trénování a testování modelu.

5.1 Sbírání dat pro tvorbu datového setu - Sniffer

Pro zachycení 802.11 rámců je zapotřebí nastavit síťovou kartu do monitorovacího režimu a spustit libovolnou formu sledování komunikace. Je možné, že systém nebude schopen zachycovat všechny rámce. V datové sadě použíté v této práci chybí část komunikace týkající se datových paketů.
Sbírání paketů bylo v této práci prováděno na stolním počítači s operačním systémem **Ubuntu 22.04 LTS** za použití nástroje *Wireshark*. Alternativně lze použít libovolný operační systém a nástroje jako *airodump-ng*, *tshark* nebo *Kismet*. Je nutno podotknout, že síťová karta **musí** podporovat **monitorovací** režim.

Nastavení síťové karty do monitorovacího režimu

Zkontrolovat, zda síťová karta režim podporuje, lze pomocí příkazu (**linux**):

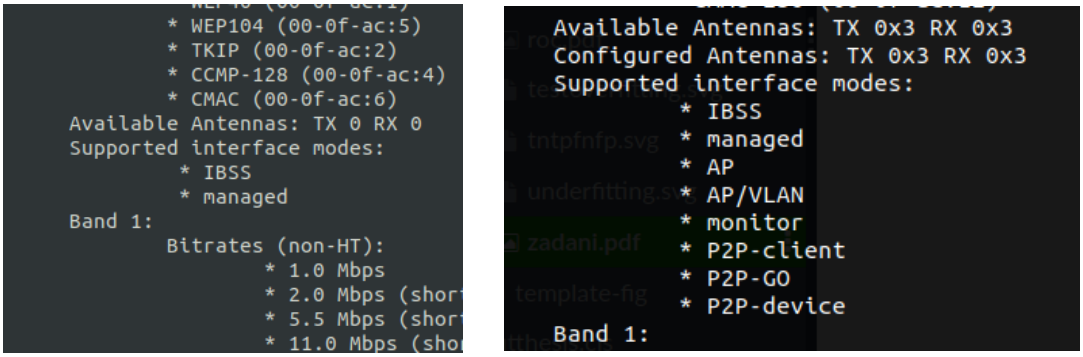
```
$ iw list
```

Síťovou kartu lze do monitorovacího režimu převést více způsoby, kupříkladu užitím nástroje *airmon-ng*. Lze však kartu přenastavit i běžnými příkazy.
Nejprve je potřeba zastavit Network Manager:

```
$ sudo systemctl stop NetworkManager
```

Poté je nutné deaktivovat bezdrátový adaptér:

```
$ sudo ifconfig INTERFACE down
```



Obrázek 5.1: Výstup příkazu $ iw list. Síťová karta v levém obrázku **nepodporuje** monitorovací režim. Síťová karta v pravém obrázku podporuje monitorovací režim.

Občas je zapotřebí ukončit zasahující procesy:

```
$ sudo airmon-ng check kill
```

Po předchozích příkaz by mělo být možné aktivovat monitorovací režim:

```
$ sudo iwconfig INTERFACE mode monitor
```

Zkontrolovat, zda je rozhraní v monitorovacím režimu lze pomocí:

```
$ iwconfig
```

Po skončení lze rozhraní nastavit do běžného režimu pomocí:

```
$ sudo iwconfig INTERFACE mode managed
$ sudo ifconfig INTERFACE up
$ sudo systemctl start NetworkManager
```

Jakmile je karta v monitorovacím režimu, lze poslouchat okolní přenos na daném kanálu. Pro zachycení komunikace byl použit nástroj *Wireshark*, ale lze použít i jiné. Celkově bylo zachyceno zhruba 2,5 hodiny provozu. Součástí sběru dat byl i sběr dat pro testování. Testovací sada obsahuje 2 anomálie, které se dále snažím rozpoznat.
První z anomálií je okolo 15 nepovedených autentizací vůči síti. Tyto nepovedené autentizace byly vytvářeny ručně pokusy o připojení z mobilu.
Druhá anomálie byla vytvořena pomocí DoS útoku na jednu z přítomných stanic. Útok byl prováděn pomocí nástroje *hping3*:

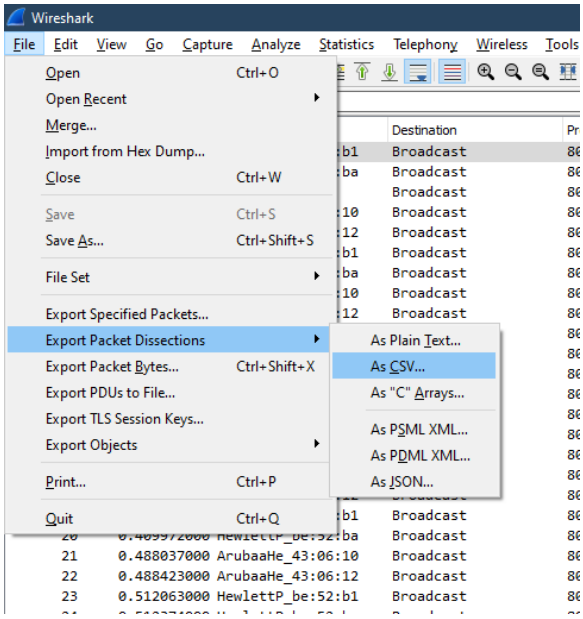
```
$ hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --flood --rand-source
```

Přestože se nejednalo o útok nativní L2, byl dost intenzivní, aby způsobil pád AP.

5.2 Extrakce informací z rámců - Extraktor

Pro extrakci byla použita funkce nástroje *Wireshark* pro exportování jen relevantních informací o jednotlivých rámcích. Vybranými sloupci byla adresa odesilatele a příjemce (source/destination address), které jsou vyhodnoceny na základě směru komunikace (to/from DS), čas zachycení rámce od počátečního spuštění, pořadí rámce, číslo fragmentu, nadtyp a podtyp rámce. V seznamu lze vidět *Wireshark* názvy display filtrů, které byly vytaženy do zobrazení:

- Time since reference or last frame - frame.time_relative
- Fragment number - wlan.frag
- Type - wlan.fc.type
- Subtype - wlan.fc.subtype

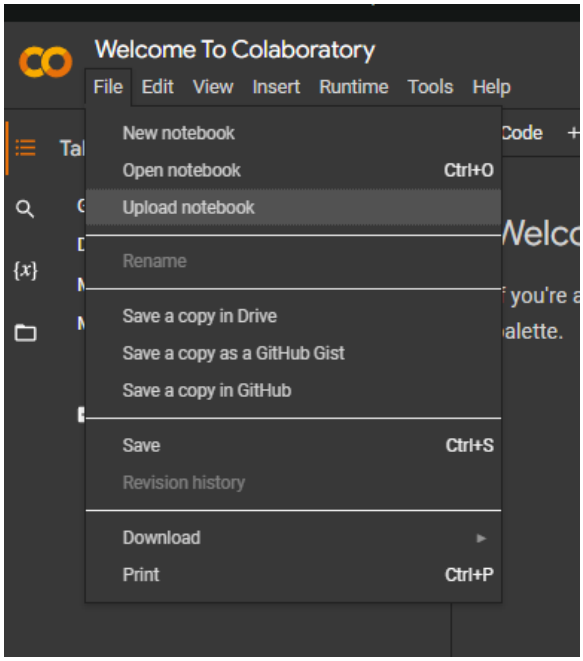


Obrázek 5.2: Exportování packet dissection pomocí *Wireshark*

Destination a Source adresy, spolu s Protocol, jsou zobrazeny v základu. V příloze B.3 je obrázek výstupu popsaného v této části.
Informace o zachycené komunikaci lze takto vyjádřit jako tabulku, kde každý řádek vyjadřuje daný rámec. *Wireshark* dovoluje tento tzv. packet dissection exportovat v CSV formátu. V obrázku 5.2 lze vidět, jak toto provést. Program přijímá dva CSV soubory *data-normal.csv* a *data-test.csv*, uložené v ZIP archívu *training.zip*.

5.3 Google Colab

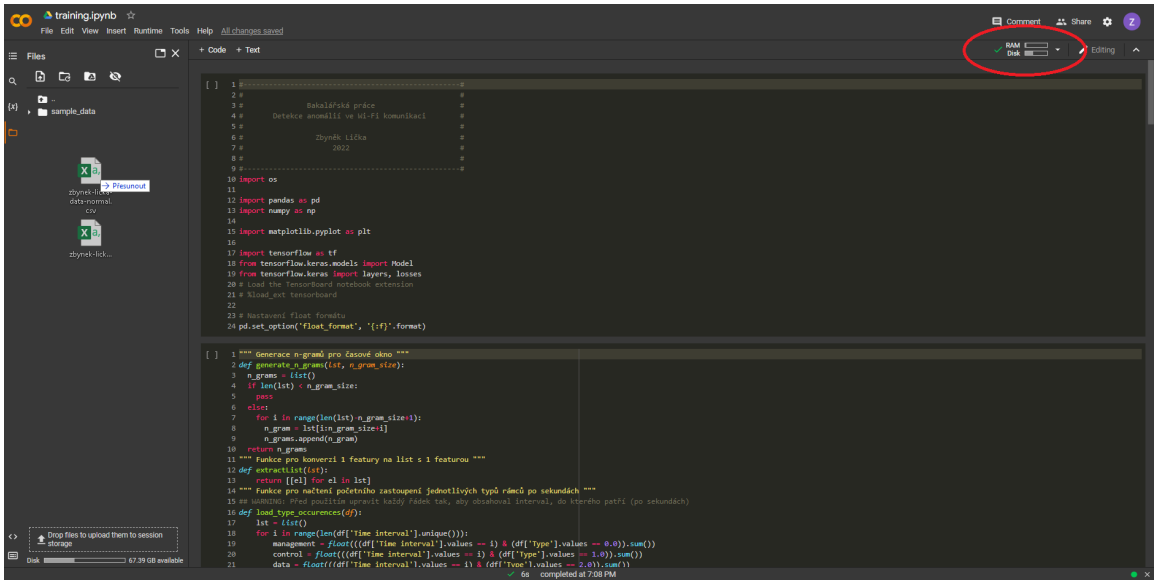
Na stránce Google Colab lze vložit *.ipynb* notebook, jak lze vidět v obrázku 5.3. Data lze vložit do repositáře vlevo po připojení k serveru (obrázek 5.4). Spustit vše lze poté pomocí Runtime -> Run all (Ctrl+F9).
Soubor Jupyter notebook *training.ipynb* obsahuje program pro trénování a vyhodnocení neuronové sítě. Do runtime je prostředí je potřeba importovat soubor *training.zip*. Program *training.ipynb* vytrénuje a vyhodnotí model za necelých 20 minut. Na konci tréninku bude v prostředí soubor *model.zip*, který obsahuje uložené modely neuronových sítí, jejich roc křivky, nastavení a výsledky detekce testovacích dat.



Obrázek 5.3: Vložení python notebooku do Google Colab

5.4 Předzpracování dat

Předzpracování dat probíhá zejména ve funkcích *generate_n_grams()*, *load_type_occurences()* a *load_data()*. Pokud je vstupní soubor pro trénování příliš dlouhý, je zapotřebí pro trénování užít generátor dat, který dokáže snížit paměťovou náročnost programu. Implementací generátoru dat je třída *TypeSequenceGenerator*.
Funkce *load_data()* nejen načítá data z CSV souborů, ale rovnou odděluje 802.11 komunikaci od ostatních protokolů (LLC, WLAN, EAPOL) a konvertuje typ rámce na decimální tvar. Dále vytváří intervaly po 1 sekundě zaokrouhlením času zachycení rámce od počátku. Rozdělení zachycené komunikace do intervalů po 1 sekundě je provedeno pomocí sloupce *Time since reference or first frame* a funkce *numpy.floor()*, která zaokrouhlí *float* hodnotu směrem dolů. Takto lze snadno spočítat množství rámců v intervalu po 1 sekundě. Tato funkce dále čistí adresy odesilatele a příjemce. Když značky to_ds a from_ds jsou obě 0, *Wireshark* vyhodnotí adresu s dodatkem, že se nacházela v TA (transmitter address) poli hlavičky 802.11. Funkce jednoduše ponechá pouze první string textu v polích adres. Pokud se v poli žádná adresa nenachází, je nahrazena hodnotou *'missing'*.
Funkce *load_type_occurences()* počítá zastoupení kategorií rámců (management, control a data) napříč zachycenými rámci v časovém rozmezí. Zastoupení počítá jako sumu rámců dané kategorie. Časové rozmezí bylo zvoleno po 1 sekundě. Funkce vrací K počtů ramců v daných intervalech. Tuto posloupnost lze dále zpracovat pomocí funkce *generate_n_grams()* nebo třídy *DataGenerator*.
Posloupnost typů rámců je jednoduše vytvořena pomocí funkce *to_list()* z knihovny *Pandas* nad sloupcem *'Frame Type'*. Tuto posloupnost taktéž lze dále zpracovávat pomocí funkce *generate_n_grams()* nebo třídy *DataGenerator*.
Funkce *generate_n_grams()* je implementací generátoru n-gramů systému. Funkce vytváří 3D list, který se skládá ze sekvencí informací. Vzdálenost (krok) mezi jednotlivými n-gramy je vždy 1. Délka sekvencí je nastavena v parametru n.



Obrázek 5.4: Připojení k runtime a vložení souborů do runtime v Google Colab

Nahrazením předchozí funkce v případě velkého množství dat je třída *DataGenerator* (navržená podle příspěvku [2]), která dědí z třídy *tf.keras.utils.Sequence* a implementuje dynamické generování batch pro model. Velikost n-gramů (délka sekvencí) je definována na začátku programu v proměnných *n_gram_size* a *time_window*. *n_gram_size* je užita pro délku okna u sekvencí typů rámců a *time_window* pro množství typů rámců v časovém rozmezí. Hodnoty proměnných jsou po řadě 128 a 60. Délka sekvencí je relativně vysoká a to z důvodu zachování kontextu. Čím delší je okno, tím více informací neuronová síť má. Vzdálenost (krok) mezi jednotlivými n-gramy je vždy 1.

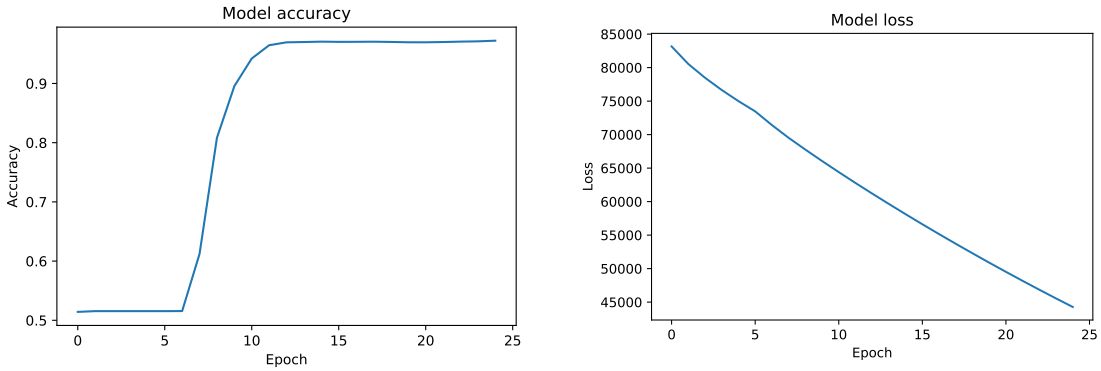
5.5 Model

```
##Classifier - neuronová síť
#!param[in] input_size Délka sekvence.
#!param[in] n_features Dimenzionalita dat - tj. počet sloupců dat.
#!warning Zkontrolovat, zda inputSize//4 je větší než 0
class Classifier(Model):
  def __init__(self, input_size, n_features):
    super(Classifier, self).__init__()

    # Vrstvy lstm
    self.classifier = tf.keras.Sequential([
      # Encoder
      layers.LSTM(input_size//2, activation='tanh', input_shape=(input_size, n_features), return_sequences=True),
      layers.LSTM(input_size//4, activation='tanh', return_sequences=True),
      # Decoder
      layers.LSTM(input_size//4, activation='tanh', return_sequences=True),
      layers.LSTM(input_size//2, activation='tanh', return_sequences=True),
      layers.TimeDistributed(layers.Dense(n_features, activation='linear'))
    ])

  def call(self, x):
    return self.classifier(x)
```

Obrázek 5.5: Implementace modelu detektoru.

Implementace modelu je v celku stručná a lze ji vidět na obrázku 5.5. Detektor dědí třídu *Model* z knihovny *Keras*. Množství LSTM buněk je dán parametricky pomocí *input_size*. V neuronové síti vzniká úzké hrdlo, které dělá rekonstrukci sekvence obtížnější. LSTM vrstvy jsou implementovány pomocí knihovny *Keras*. Model vrací rekonstruovanou sekvenci dat, pomocí které lze vypočítat rekonstrukční chybu, která je dále použíta při trénování či detekci.

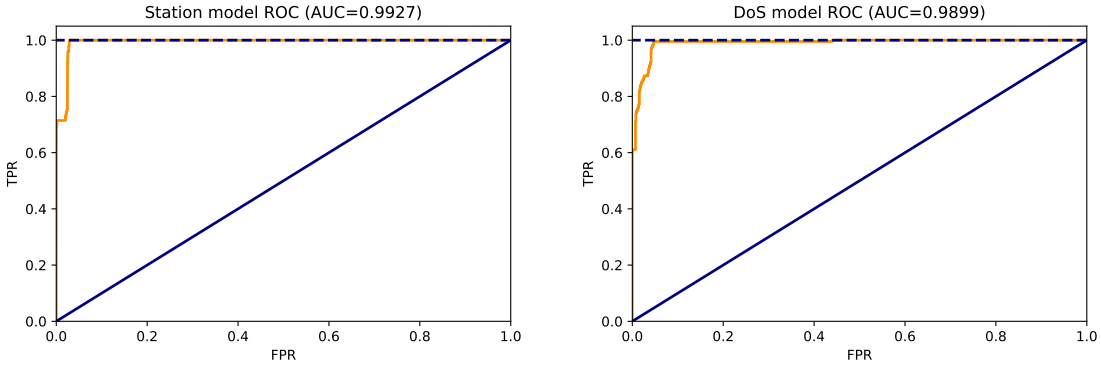
5.6 Trénování a testování modelu



Obrázek 5.6: Trénovací loss a přesnost pro model *dos_model*.

Program *trainig.ipynb* implementuje proces trénovaní modelu. Model lze trénovat pomocí funkce *self.fit()* implementované knihovnou *Keras*. Detektor používá dva modely *dos_model*, který modeluje posloupnosti množství typů rámců, a *station_model*, který modeluje typickou komunikaci stanice vůči AP pomocí posloupností typů rámců.Předzpracovaná data jsou použita pro tvorbu sekvencí, které jsou následně použity pro trénování modelu. Sekvence jsou tvořeny funkcí *generate_n_grams()*, ale v případě příliš velkého datového setu lze použít i generátor *DataGenerator*.

Množství epoch, které udává, kolikrát bude model trénován na stejných datech, je v případě



Obrázek 5.7: ROC křivky modelů (*station_model* vlevo, *dos_model* vpravo).

dos_model 25 a v případě *station_model* 4. V druhém případě je model trénován na datech více stanic vůči AP a pro každou stanici je trénován 4krát. Počet zvolených stanic je 3 a

dané stanice byly zvoleny, protože byly známé, a bylo jisté, že komunikují standartně. Stanice jsou během tréninku vybírány náhodně. U modelu *dos_model* bylo zvoleno množství epoch experimentálně. V obrázku 5.6 je vidět, že po 25 epochách přesnost na trénovacích datech konverguje.

Velikost batch byla zvolena 32 pro *dos_model* a 64 pro *station_model*. Velikost batch se odvíjí od množství dat a byla zvolena experimentálně.

Jako optimalizátor byl zvolen *ADAM* s mírou učení 0,001 v případě *dos_model* a 0.0006 v případě *station_model*. Míra učení byla zvolena experimentálně. Pro výpočet chyby je použit MSE (Mean Squared Error). ADAM i MSE je implementováno knihovnou *Keras*.

Oproti *station_model*, *dos_model* vykazuje vysokou chybu i na konci trénování. Při ověřování modelu bylo zjištěno, že toto nevylučuje schopnost detekovat.

Testování probíhá nad souborem *data-test.csv*. Nad sekvencemi generovanými z testovacích dat je provedena detekce pomocí rekonstrukční chyby příslušných modelů. ROC křivky jednotlivých modelů jsou vytvořeny porovnáním rekonstrukčních chyb s manuálně označenými daty. ROC a AUC (Area Under the Curve) obou modelů lze vidět v 5.7. Detekce je provedena pomocí funkce *self.predict()* a rekonstručkní chyby jsou vypočítané pomocí funkce *keras.losses.mse()* (obojí implementace knihovny *Keras*). Aby výpočet chyby byl pro celou sekvenci, je třeba konvertovat výsledek detekce a vstupní data na 2D pole. Toto je provedeno pomocí funkce *self.reshape()* po konvertování listu na *numpy array*. Vypočtená rekonstrukční chyba je zobrazena do grafu a graf uložen jako PDF pomocí knihovny *Matplotlib*. Hodnoty ROC křivky jsou vypočítány pomocí funkce *roc_curve()* z knihovny *scikit-learn* a vizualizovány pomocí knihovny *Matplotlib*. Hodnota AUC byla vypočítána pomocí funkce *roc_auc_score()* z knihovny *scikit-learn*.

Vytrénované modely jsou uloženy do složek *log/dos_model* a *log/station_model*. Na konci trénovacího procesu jsou modely a výsledky testování uloženy do ZIP archívu *model.zip*. Do archívu je taky uloženo nastavení ve formátu YAML. Nastavení obsahuje vybrané velikosti n-gramů pro oba modely, MAC adresu AP, vůči kterému byl model trénován, a známé stanice, kterými byl trénován model *station_model*.

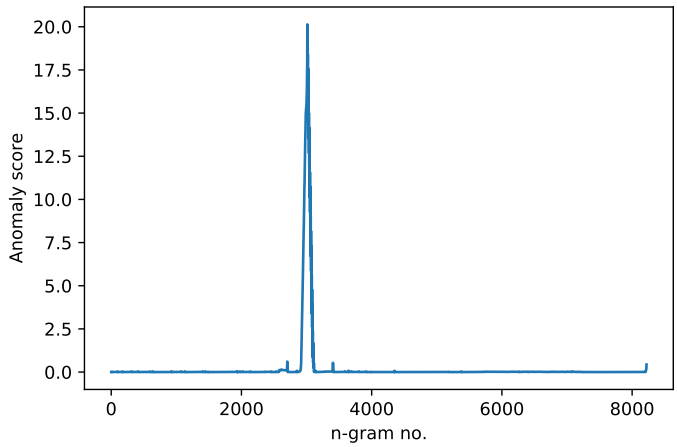
Kapitola 6

Experimenty a diskuze

Tato kapitola se bude zabývat popisem dvou experimentů, jejich výsledků a vyhodnocení. Na konci této kapitoly lze také najít diskuzní část, kde jsou vypíchnuty známé nedostatky systému a návrhy k rozšíření.

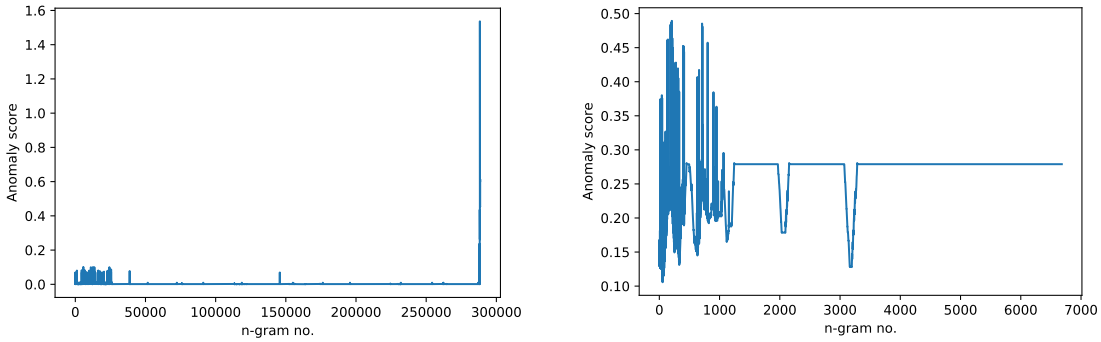
Pro experimentování byly zvoleny dvě anomálie z nichž první se skládá z hádání hesla skutečným uživatelem a druhá *hping3* DoS útokem na jedno ze zařízení v síti. Cílem prvního experimentu je zkontrolovat citlivost systému. Druhý experiment kontroluje schopnost systému vypořádat se s DoS útokem. Pomocí těchto dvou experimentů jsou vytvořeny ROC křivky modelů.

6.1 Hádání hesla



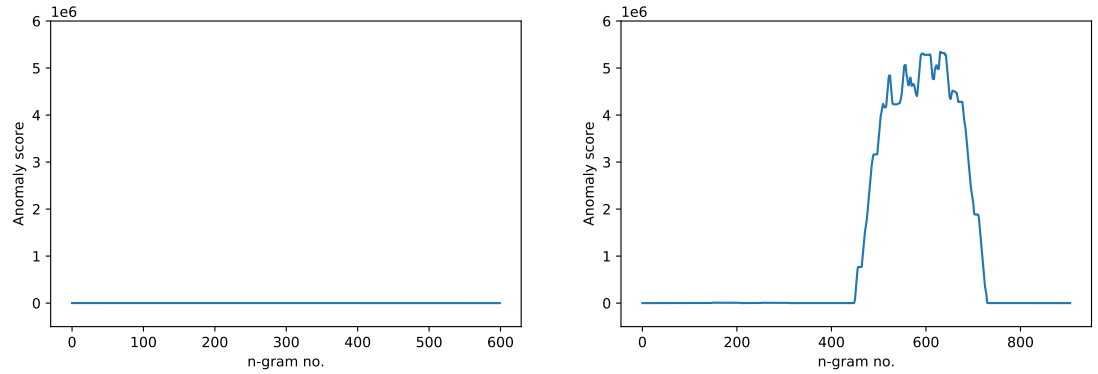
Obrázek 6.1: Graf anomaly score pro hádání hesla. Rekonstrukční chyba modelu *station_model*.

Anomálie Uživatel odpojený od sítě se pokusil 25krát nepovedeně připojit k síti. Uživatel opakovaně ručně zadával špatné heslo dokud se konečně nepřipojil po řadě pokusů.



Obrázek 6.2: Detekce anomálií nad posloupností typů rámců ostatních stanic v komunikaci.

Detekce Výsledek detekce lze vidět na obrázku 6.1. Komunikace zařízení, ze kterého bylo prováděno hádání hesla, v polovině vykazuje vysokou anomalitu. Sekvence vykazující vysokou anomalitu obsahují zvýšený počet autentikačních rámců. Detekce útoku zapadá do rozmezí skutečného útoku. Na obrázku 6.2 lze vidět detekci anomálií u ostatních stanic. Obrázek vlevo na konci komunikace projevuje vysokou anomálnost. V této oblasti se také vyskytuje zvýšený počet autentikačních rámců.



Obrázek 6.3: Graf anomálnosti před a během DoS útoku. Rekonstrukční chyba modelu *dos_model*.

Anomálie Pomocí nástroje *hping3* byl na jedno ze zařízení v síti vyslán DoS útok.

Detekce Výsledek detekce lze vidět na obrázcích 6.3. Komunikace v čas útoku začne vykazovat zvýšující se anomálnost. Po zvětšení délky sekvence došlo k vyhlazení výstupu. Nutné podotknout, že se nejedná o pouhé zatížení sítě, ale výsledek ohodnocení modelu, tedy anomálnost komunikace. Rekonstrukční chybu při útoku lze dát do kontrastu s rekonstručkní chybou během běžného provozu s absencí anomálií před útokem. Zvýšená anomálnost zapadá do rozmezí skutečného útoku.

6.2 Vyhodnocení

Systém byl schopen rozpoznat obě známé anomálie.
Model *station_model* se zdá být citlivý na posloupnosti obsahující typy rámců, které by

se v rámci trénovacích dat daly považovat za odlehlé hodnoty (outliery). Pokud se takové hodnoty v rámci jedné sekvence objeví vícekrát, měla by vyskočit anomálnost této sekvence, protože tyto hodnoty model špatně rekonstruuje.
Model *dos_model* obecně dosahuje dobrých výsledků.

6.2.1 Diskuze

Výsledný systém je schopen odhalit hádání hesla hlavně proto, že je autentikačních rámců samotných v datech málo. Model pro sekvence typů rámců se může jevit zbytečný, ale je snadno rozšiřitelný a zachovává kontext v rámci posloupnosti samotných rámců, který může být důležitý pro detekci anomálií.
Model pro množství typů rámců se zdá být dosti efektivní. Pro tento model by se daly použít všechny typy rámců, ale při velkém množství dimenzí by se číslně malé, avšak těžce anomální, změny nemusely dostatečně projevit.
Byl proveden pokus o rozšíření detekce na běh v reálném čase pomocí knihovny *pyshark*, avšak zpracování rámců nebylo dostatečně rychlé. Pokud by měla být detekce takto rozšířena (pravděpodobně se zpožděním), bylo by vhodné ji implementovat pomocí jazyka C++. Knihovna *TensorFlow* má implementaci pro C++ a pro zachycení rámců by šlo použít knihovny *libpcap*.
Práce se záměrně zabývá pouze 802.11 komunikací, ale některé z útoků používají ARP, které nejsou přímou součástí standardu. Bylo by vhodné přidat ARP do detekce zastoupení rámců. Bylo by také vhodné sledovat množství fragmentů v komunikaci, kvůli fragmentačním útokům. Systém také nevyužívá EAPOL a LLC rámce, které by také mohly být zahrnuty v procesu detekce.

Kapitola 7

Závěr

V této práci jsem se seznámil se způsoby detekce anomálií v prostředí počítačových sítí. Analyzoval jsem požadavky na systém ve formě analýzy běžných útoků na standard IEEE 802.11 a charakteru IEEE 802.11 standardu. Navrhl a naimplementoval jsem systém pro detekci anomálií ve Wi-Fi komunikaci použitím metod umělé inteligence, konkrétně LSTM rekurzivních neuronových sítí užitím učení bez učitele. Systém jsem ověřil na realných datech skutečného provozu a skutečných útoků. Výsledný systém vykazuje schopnost detekovat jak DoS útoky, tak méně čitelné primitivní útoky. AUC výsledných modelů jsou 0,9899 a 0,9768. Jejich ROC křivky lze vidět na obrázku 5.7. Model pro posloupnosti typů rámců se zdá být citlivý na posloupnosti obsahující typy rámců, které by se v kontextu trénovacích dat daly považovat za odlehlé hodnoty (outliery). Zmíněny model se projevil efektivní v detekování útoku, který se značil zvýšeným počtem autentikačních rámců v krátkém období. Model pro zastoupení typů rámců dosáhl dobrých výsledků. Při detekci DoS útoku se projevil efektivní a bez problémů. Na konci práce jsem diskutoval výsledky a možnosti rozšíření.

Literatura

[1] Alipour, H., Al Nashif, Y. B., Satam, P. a Hariri, S. Wireless Anomaly Detection Based on IEEE 802.11 Behavior Analysis. *IEEE Transactions on Information Forensics and Security* [online]. [New York]: IEEE. October 2015, sv. 10, č. 10, s. 2158–2170, [cit. 2022-07-03]. DOI: 10.1109/TIFS.2015.2433898. ISSN 1556-6021. Dostupné z: https://ieeexplore.ieee.org/document/7109166.

[2] Amidi, A. a Amidi, S. A detailed example of how to use data generators with Keras. *Afshine Amidi and Shervine Amidi Blog* [online], 19. May 2018 [cit. 2022-7-2]. Dostupné z: https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly.

[3] Buczak, A. L. a Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys Tutorials* [online]. [New York]: IEEE. 2016, sv. 18, č. 2, s. 1153–1176, [cit. 2022-01-31]. DOI: 10.1109/COMST.2015.2494502. ISSN 1553-877X. Dostupné z: https://ieeexplore.ieee.org/document/7307098.

[4] Chandola, V., Banerjee, A. a Kumar, V. Anomaly Detection: A Survey. *ACM Computing Surveys* [online]. New York, NY, USA: Association for Computing Machinery. July 2009, sv. 41, č. 3, s. 58, [cit. 2022-07-08]. DOI: 10.1145/1541880.1541882. ISSN 0360-0300. Dostupné z: https://doi.org/10.1145/1541880.1541882.

[5] Darchis, N. 802.11 frames: A starter guide to learn wireless sniffer traces. *Cisco Community* [online], 25. října 2010. 02-03-2022 [cit. 2022-07-02]. Dostupné z: https://community.cisco.com/t5/wireless-mobility-documents/802-11-frames-a-starter-guide-to-learn-wireless-sniffer-traces/ta-p/3110019.

[6] El Khatib, K. Impact of Feature Reduction on the Efficiency of Wireless Intrusion Detection Systems. *IEEE Transactions on Parallel and Distributed Systems* [online]. [New York]: IEEE. August 2010, sv. 21, č. 8, s. 1143–1149, [cit. 2022-07-05]. DOI: 10.1109/TPDS.2009.142. ISSN 1558-2183. Dostupné z: https://ieeexplore.ieee.org/document/5226620.

[7] Eltanbouly, S., Bashendy, M., AlNaimi, N., Chkirbene, Z. a Erbad, A. Machine Learning Techniques for Network Anomaly Detection: A Survey. In: IEEE. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)* [online]. Doha (Qatar): IEEE, May 2020, s. 156–162 [cit. 2022-01-31]. DOI: 10.1109/ICIoT48696.2020.9089465. ISBN 978-1-7281-4821-2. Dostupné z: https://ieeexplore.ieee.org/document/9089465.

[8] Fehér, D. J. a Sandor, B. Effects of the WPA2 KRACK Attack in Real Environment. In: IEEE. *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)* [online]. Subotica (Serbia): IEEE, November 2018, s. 000239–000242 [cit. 2022-6-30]. DOI: 10.1109/SISY.2018.8524769. ISBN 978-1-5386-6841-2. Dostupné z: https://ieeexplore.ieee.org/document/8524769.

[9] Fluhrer, S., Mantin, I. a Shamir, A. Weakness in the Key Scheduling Algorithm of RC4. In: Vaudenay, S. a Youssef, A. M., ed. *Selected Areas in Cryptography* [online]. Berlin, Heidelberg: Springer, December 2001, sv. 2259, s. 1–24 [cit. 2022-6-28]. DOI: 10.1007/3-540-45537-X_1. ISBN 978-3-540-45537-0. Dostupné z: link.springer.com/chapter/10.1007/3-540-45537-X_1.

[10] Goodfellow, I., Bengio, Y. a Courville, A. *Deep Learning* [online]. [Cambridge (Massachusetts, USA)]: MIT Press, 2016 [cit. 2022-01-31]. Knížka má ISBN, ale byly použity volně dostupné materiály na uvedené stránce. Dostupné z: http://www.deeplearningbook.org.

[11] Grubbs, F. E. Procedures for Detecting Outlying Observations in Samples. *Technometrics* [online]. [London (Velká Británie)]: Informa UK Limited. February 1969, sv. 11, č. 1, s. 1–21, [cit. 2022-07-08]. DOI: 10.1080/00401706.1969.10490657. Dostupné z: https://doi.org/10.1080%2F00401706.1969.10490657.

[12] Heckerman, D. A Tutorial on Learning with Bayesian Networks. In: Jordan, M. I., ed. *Learning in Graphical Models* [online]. Dordrecht: Springer Netherlands, 1998, s. 301–354 [cit. 2022-01-31]. NATO ASI Series, č. 89. DOI: 10.1007/978-94-011-5014-9_11. ISBN 978-94-011-5014-9. Dostupné z: https://doi.org/10.1007/978-94-011-5014-9_11.

[13] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE Std 802.11-1997* [online]. [New York]: IEEE. November 1997, s. 445, [cit. 2022-07-27]. DOI: 10.1109/IEEESTD.1997.85951. Dostupné z: https://ieeexplore.ieee.org/servlet/opac?punumber=5258.

[14] IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements. *IEEE Std 802.11i-2004* [online]. [New York]: IEEE. July 2004, s. 190, [cit. 2022-6-29]. DOI: 10.1109/IEEESTD.2004.94585. Dostupné z: https://ieeexplore.ieee.org/servlet/opac?punumber=9214.

[15] Kolias, C., Kambourakis, G., Stavrou, A. a Gritzalis, S. Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Communications Surveys & Tutorials* [online]. [New York]: IEEE. 2016, sv. 18, č. 1, s. 184–208, [cit. 2022-06-28]. DOI: 10.1109/COMST.2015.2402161. ISSN 1553-877X. Dostupné z: https://ieeexplore.ieee.org/document/7041170.

[16] Kurose, J. F. a Ross, K. W. *Computer Networking: A Top-Down Approach*. 7. vyd. Boston: Pearson Education, 2017. ISBN 978-0-13-359414-0.

[17] Lu, W. a Traore, I. Detecting New Forms of Network Intrusion Using Genetic Programming. *Computational Intelligence* [online]. [New Jersey (USA)]: Wiley. July

2004, sv. 20, č. 3, s. 475–494, [cit. 2022-01-31]. DOI: 10.1111/j.0824-7935.2004.00247.x. ISSN 1467-8640. Dostupné z: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0824-7935.2004.00247.x.

[18] Sasaki, Y. The truth of the F-measure. *Teach Tutor Mater* [online]. 26th October, 2007. Manchester (Velká Británie): School of Computer Science, University of Manchester. Leden 2007, s. 5. Dostupné z: https://www.researchgate.net/publication/268185911_The_truth_of_the_F-measure.

[19] Satam, P. a Hariri, S. WIDS: An Anomaly Based Intrusion Detection System for Wi-Fi (IEEE 802.11) Protocol. *IEEE Transactions on Network and Service Management* [online]. [New York]: IEEE. March 2021, sv. 18, č. 1, s. 1077–1091, [cit. 2022-07-03]. DOI: 10.1109/TNSM.2020.3036138. ISSN 1932-4537. Dostupné z: https://ieeexplore.ieee.org/document/9249426.

[20] Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena* [online]. [Amsterdam]: Elsevier. March 2020, sv. 404, s. 28, [cit. 2022-06-26]. DOI: https://doi.org/10.1016/j.physd.2019.132306. ISSN 0167-2789. Číslo článku 132306. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0167278919305974.

[21] Stallings, W. *Network Security Essentials*. 4. vyd. Upper Saddle River, New Jersey USA): Pearson Education, 2010. ISBN 978-0-13-610805-4.

[22] Tharwat, A. Classification assessment methods. *Applied Computing and Informatics* [online]. [Bingley (Velká Británie)]: Emerald Publishing Limited. Leden 2021, sv. 17, č. 1, s. 168–192, [cit. 2020-07-15]. DOI: 10.1016/j.aci.2018.08.003. ISSN 2634-1964. Dostupné z: https://doi.org/10.1016/j.aci.2018.08.003.

[23] Vanhoef, M. a Piessens, F. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. In: ACM. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* [online]. New York, NY, USA: Association for Computing Machinery, October 2017, s. 1313–1328 [cit. 2022-06-30]. CCS '17. DOI: 10.1145/3133956.3134027. ISBN 9781450349468. Dostupné z: https://doi.org/10.1145/3133956.3134027.

[24] Vanhoef, M. a Ronen, E. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. In: IEEE. *2020 IEEE Symposium on Security and Privacy (SP)* [online]. San Francisco, CA, USA: IEEE, July 2020, s. 517–533 [cit. 2022-06-30]. DOI: 10.1109/SP40000.2020.00031. ISBN 978-1-7281-3497-0. Dostupné z: https://ieeexplore.ieee.org/document/9152782.

Seznam příloh

Příloha A

Obsah přiloženého paměťového média

- Adresář *licenses* obsahuje licenci Python a použitých knihoven.
- Pcap soubory *data-hping.pcap* a *data-normal.pcap* obsahují zachycenou komunikaci použitou pro trénování a testování modelu.
- PDF soubor *thesis.pdf* obsahuje text práce.
- Soubor *manual.txt* obsahuje stručné instrukce, jak zprovoznit program.
- Python notebook *training.ipynb* obsahuje implementaci systému navrženého v této práci.
- Archív *training.zip* obsahuje CSV exportaci packet slice z nástroje *Wireshark*.
- Archív *thesis.zip* obsahuje zdrojový tvar písemné zprávy.

Příloha B

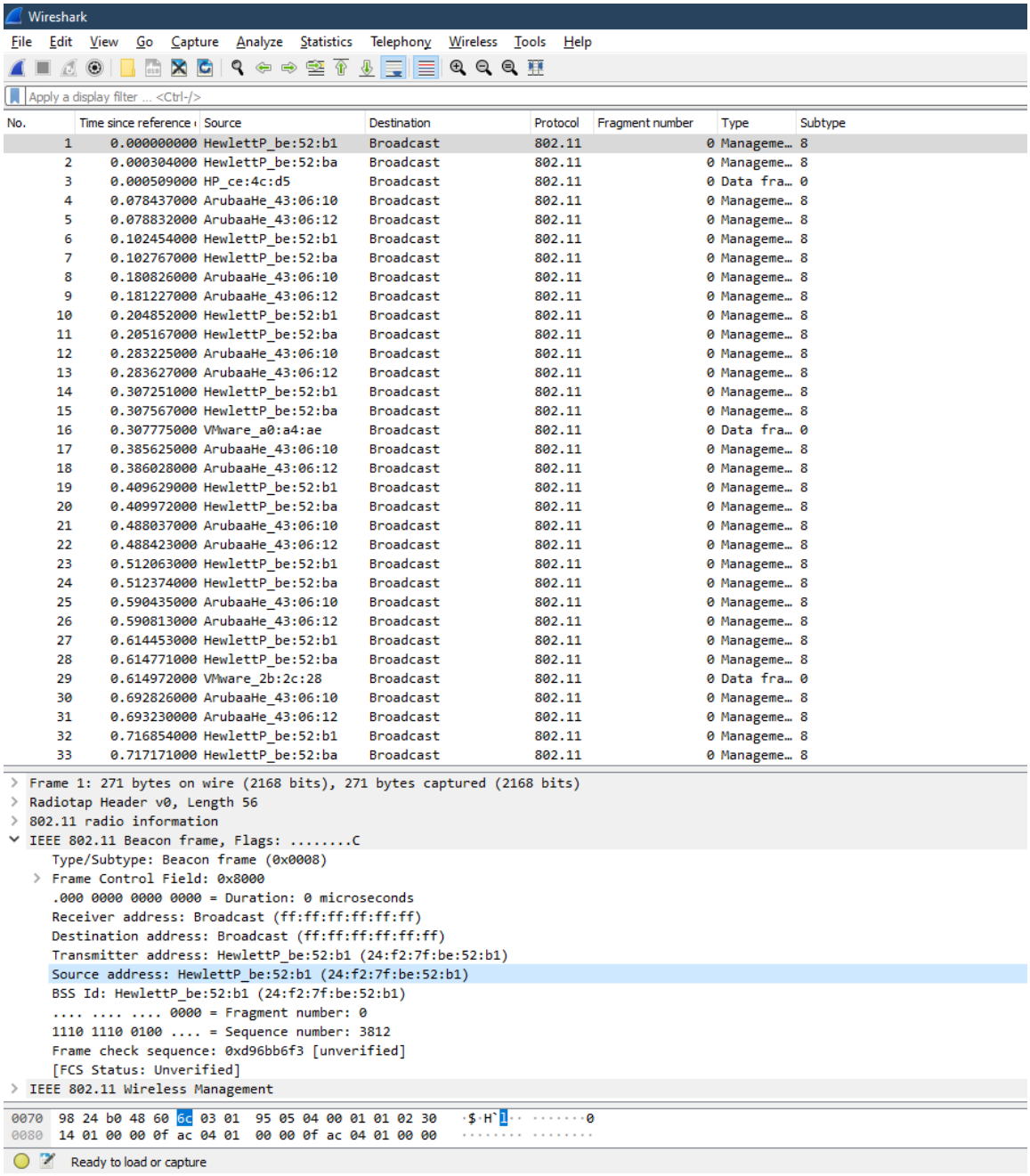
IEEE 802.11

Typ rámce	Dec.hod.	Hex. hod.	Nadtyp rámce
Association Request	0	0x00	Management
Association Response	1	0x01	
Reassociation Request	2	0x02	
Reassociation Response	3	0x03	
Probe Request	4	0x04	
Probe Response	5	0x05	
Beacon	8	0x08	
ATIM	9	0x09	
Disassociation	10	0x0A	
Authentication	11	0x0B	
Deauthentication	12	0x0C	
Action	13	0x0D	
Block Ack Request	24	0x18	Control
Block Ack	25	0x19	
PS-Poll	26	0x1A	
RTS	27	0x1B	
CTS	28	0x1C	
ACK	29	0x1D	
CF-end	30	0x1E	
CF-end + CF-ack	31	0x1F	
Data	32	0x20	Data
Data + CF-ack	33	0x21	
Data + CF-poll	34	0x22	
Data +CF-ack +CF-poll	35	0x23	
Null	36	0x24	
CF-ack	37	0x25	
CF-poll	38	0x26	
CF-ack +CF-poll	39	0x27	
QoS data	40	0x28	
QoS data + CF-ack	41	0x29	
QoS data + CF-poll	42	0x2A	
QoS data + CF-ack + CF-poll	43	0x2B	
QoS Null	44	0x2C	
QoS + CF-poll (no data)	46	0x2E	
Qos + CF-ack (no data)	47	0x2F	

Obrázek B.1:

Attack	Effect	Traffic Injected	Version	Difficulty	Comments	Threat
FMS	Secret Key Cracking	>2,000,000	WEP	Easy	Slow	★★
Korek	Secret Key Cracking	>700,000	WEP	Easy	Slow	★★
PTW	Secret Key Cracking	>50,000	WEP	Easy	Fast Requires resources	★★★
Dictionary	Secret Key Cracking	1	WPA/WPA2	Easy	depends on weak passwords	★★
Chopchop	Keystream Retrieval Packet Decryption Keystream Retrieval	<=256*n	WEP	Moderate	-	★★
Fragmentation	Packet Decryption	<=16	WEP	Moderate	Reveals up to 64 Slow	★★
Caffe Latte	Secret Key Cracking without AP	<=65280	WEP	Easy	not possible against all OSs	★
Hitre	Secret Key Cracking without AP	1	WEP	Easy	Fast	★★★
Deauth	Loss of Connectivity	High	All	Easy	Can Target Client	★★★
Disassociation	Loss of Connectivity	High	All	Easy	Can Target Client	★★★
Deauth Broadcast	Loss of Connectivity	High	All	Easy	Affects All	★★★
Disassociation Broadcast	Loss of Connectivity	High	All	Easy	Affects All	★★★
Block Ack	Annoyance	Low	802.11n	High	Requires Accuracy	★
Authentication Request	Inability to join the network	High	All	Low	Ineffective Against Most Devices	★
Fake PS	Annoyance	High	All	High	Requires Accuracy	★
CTS Flooding	Annoyance	High	All	Low	Can Target Client	★
RTS Flooding	Annoyance	High	All	Low	Can Target Client	★
Beacon Flooding	Inability to join the network	High	All	Low	Effective Against Limited Devices	★
Probe Request	Annoyance	High	All	Low	Affects All	★★
Probe Response	Annoyance	High	All	Low	Can Target Client	★★
Honeypot	Loss of Privacy	None in the Network	All	Moderate	Relies on Naive Users	★★★
Evil Twin	Loss of Privacy		All	Moderate	Requires Knowledge of Secret Key	★★★
Rogue AP	Loss of Privacy	None in the Network	All	Moderate	Requires Access to the Wired Network	★★★

Obrázek B.2:


Wireshark
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time since reference Source Destination Protocol Fragment number Type Subtype
1 0.000000000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
2 0.000304000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
3 0.000509000 HP_ce:4c:d5 Broadcast 802.11 0 Data fra… 0
4 0.078437000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
5 0.078832000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
6 0.102454000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
7 0.102767000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
8 0.180826000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
9 0.181227000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
10 0.204852000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
11 0.205167000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
12 0.283225000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
13 0.283627000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
14 0.307251000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
15 0.307567000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
16 0.307775000 VMware_a0:a4:ae Broadcast 802.11 0 Data fra… 0
17 0.385625000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
18 0.386028000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
19 0.409629000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
20 0.409972000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
21 0.488037000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
22 0.488423000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
23 0.512063000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
24 0.512374000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
25 0.590435000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
26 0.590813000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
27 0.614453000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
28 0.614771000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
29 0.614972000 VMware_2b:2c:28 Broadcast 802.11 0 Data fra… 0
30 0.692826000 ArubaaHe_43:06:10 Broadcast 802.11 0 Manageme… 8
31 0.693230000 ArubaaHe_43:06:12 Broadcast 802.11 0 Manageme… 8
32 0.716854000 HewlettP_be:52:b1 Broadcast 802.11 0 Manageme… 8
33 0.717171000 HewlettP_be:52:ba Broadcast 802.11 0 Manageme… 8
Frame 1: 271 bytes on wire (2168 bits), 271 bytes captured (2168 bits)
Radiotap Header v0, Length 56
802.11 radio information
IEEE 802.11 Beacon frame, Flags: C
Type/Subtype: Beacon frame (0x0008)
Frame Control Field: 0x8000
.000 0000 0000 0000 = Duration: 0 microseconds
Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
Transmitter address: HewlettP_be:52:b1 (24:f2:7f:be:52:b1)
Source address: HewlettP_be:52:b1 (24:f2:7f:be:52:b1)
BSS Id: HewlettP_be:52:b1 (24:f2:7f:be:52:b1)
.... 0000 = Fragment number: 0
1110 1110 0100 .... = Sequence number: 3812
Frame check sequence: 0xd96bb6f3 [unverified]
[FCS Status: Unverified]
IEEE 802.11 Wireless Management
0070 98 24 b0 48 60 6c 03 01 95 05 04 00 01 01 02 30
0080 14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00
Ready to load or capture

Obrázek B.3: