VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Projekt - Detekce anomálií v síťové komunikaci

Přenos dat, počítačové sítě a protokoly

Bc. Filip Weigel (xweige01)

20. dubna 2022

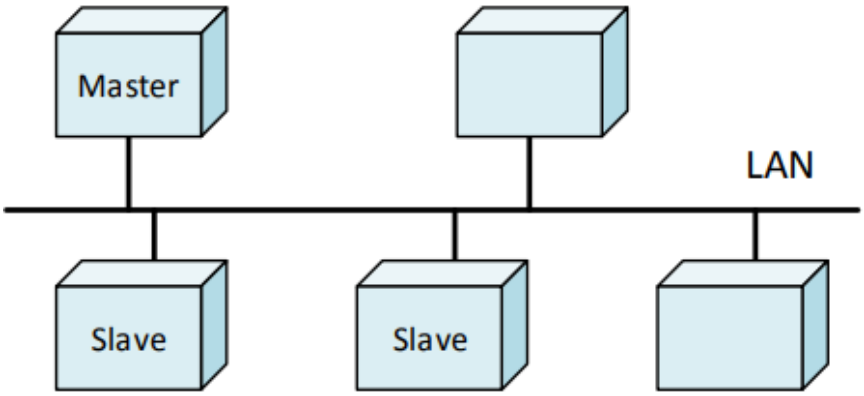
Obsah

1 Úvod

Hlavní náplní projektu je detekce anomálií v ICS komunikaci pomocí zvolených analytických metod. Za anomálii se dá považovat jakákoliv odchylka od normálu. Anomálie v počítačové komunikaci mohou plynout z různých zdrojů. Může se jednat o výpadek linky, softwarovou/hardwarovou chybou, interference v podobě elektromagnetického záření mající za vliv poškozené pakety, nebo se může jednat o počítačový útok s jakýmkoliv cílem, například získat neoprávněný přístup k majetku, ukrást majetek, škodit krátkodobě/dlouhodobě, variant je nesčetně mnoho[4][7].

Běžná síťová komunikace je nepředvídatelná. Příkladem komunikace může být uživatel prohlížející webovou stránku. Ve valné většině případů nelze předpovědět uživatelovo chování. Bude pokračovat v prohlížení současné stránky, nebo bude vyhledávat zcela odlišnou stránku, nebo v prohlížení webu skončí a odejde od svého zařízení?

Dle [2] systémy ICS (Industrial Control System) zasílají kontrolní a monitorovací data mezi zařízeními v průmyslovém prostředí. Jako příklad lze uvést rozvody plynu, vody, nebo elektřiny. Komunikace v sítích ICS je periodická a stabilní (deterministická), což je velký rozdíl oproti komunikaci běžné. Při přenosu protokolu IEC 104 kombinuje aplikační vrstvu protokolu IEC 60870-5-101 a transportní funkcionalitu, kterou poskytuje protokol TCP/IP. IEC 60870-5-101 protokol poskytuje komunikační profil k zasílání tele-kontrolních zpráv mezi centrální tele-kontrolující stanicí (master) a vzdálenou tele-kontrolní stanicí (slave) [3].



Obrázek 1.1: Topologie sítě Master-Slave[1]

[1] https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FPDS-IT%2Fprojects%2Frelated-papers%2FDescription+and+analysis+of+IEC+104+Protocol.pdf&cid=14774

Sledováním různých vlastností ICS komunikace, kterým může být směr komunikace, velikost paketu, inter-arrival time atd. lze vytvořit statistický model této komunikace k detekci anomálií. Příčinou jsou nestandardní síťové přenosy, selhání zařízení, výpadek linky, ale taktéž se může jednat o různé kybernetické útoky jako Denial of Service(DoS), injektování paketů atp.

Postup při vypracovávání projektu byl následující:

- Seznámení se s protokolem IEC 104
- Analýza datasetu a výběr parametrů včetně očistění dat
- Sestavení modelu s použitím statistických atributů
- Vyhodnocení modelu na základě experimentů

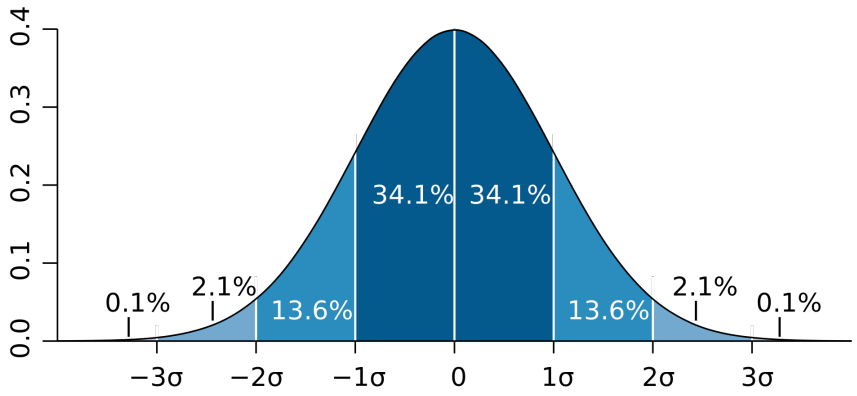
V části 2. budou prezentovány metody pro detekci anomálií, v části 3. bude prezentována implementace a v 4. části budou prezentovány následné experimenty.

2 Detekce anomálií

Statistická detekce anomálií vychází z předpokladu, že normální (bez anomálií) komunikace se vyskytuje v oblastech s vysokou pravděpodobností, zatímco anomálie se vyskytují v oblastech s nízkou pravděpodobností. Statistické modelování se těší velké oblibě, jelikož lze poměrně jednoduše a rychle detekovat odlehlé hodnoty (outliers). Hlavním předpokladem je, že zachycené body v jednorozměrném prostoru jsou rozmístěny dle určitého rozložení (Gaussovo rozdělení, Exponenciální rozložení, atp.) [2].

Pravidlo tří sigma

Pravidlo tří sigma, předpokládá že u normálně [6] rozděleného souboru by měly všechny relevantní hodnoty nacházet do 3 směrodatných odchylek (3σ) od aritmetického průměru. V okolí 1σ od průměru by se mělo nacházet přibližně 68.27% hodnot, pro 2σ zhruba 95.45% a pro 3σ 99.73% hodnot. Při zvětšování násobku směrodatné odchylky dochází k zvýšení tolerance k odlehlým hodnotám [8].



Obrázek 2.1: Graf Gaussova rozdělení[1]

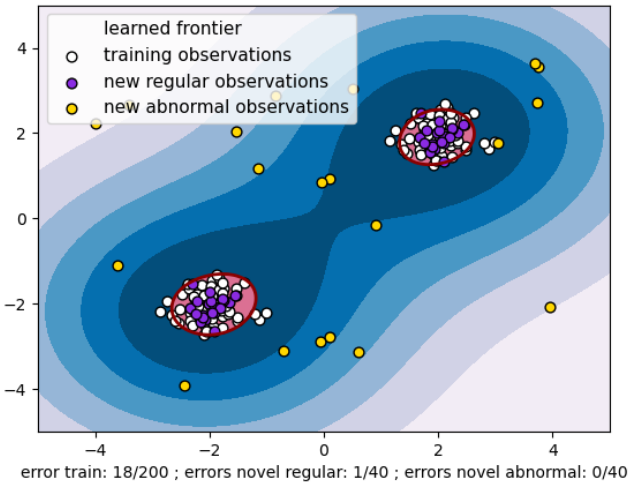
[1] https://upload.wikimedia.org/wikipedia/commons/8/8c/Standard_deviation_diagram.svg

Boxplot

Další technikou pro detekci anomálií je krabicový graf/diagram (boxplot). Jedná se o způsob grafické vizualizace dat pomocí kvartilů. Střední část diagramu je ohraničena 3. kvartilem shora a 1. kvartilem zdola. V střední části se rovněž vyskytuje linie označující medián. [5] Normální hodnoty se vyskytují v intervalu $\langle Q_1 - 1.5 * IRQ, Q_3 + 1.5 * IRQ \rangle$, kde IRQ je $Q_3 - Q_1$ [2]. Boxploty mohou (nemusí) obsahovat taktéž linie vycházející ze střední části diagramu kolmo nahoru a dolů, které vyjadřují rozmezí dat pod prvním a nad třetím kvartilem. Odlehlé hodnoty (ourliers) mohou být poté vyobrazeny jako jednotlivé body v diagramu [5].

Support vector machines

Metoda podpůrných vektorů je jedna z metod strojového učení s učitelem. Využívána je hlavně pro klasifikaci a regresní analýzu. Stavebním kamenem SVM je lineární klasifikátor do dvou tříd. Cílem je nalézt nadrovinu, která je schopna optimálně rozdělovat trénovací data do patřičných poloprostorů. Optimální podrovina je taková podrovina, že hodnota minima vzdáleností bodů od roviny je co největší (okolo nadroviny je na obě strany co nejširší pruh bez bodů, který se nazývá hraniční pásmo - maximal margin). Na popis nadroviny postačují body ležící na okraji hraničního pásma, kterých je obvykle málo. Body se nazývají podpůrné vektory (support vectors) a i název metody.
Metody SVM mají mnoho variant. Jednou z jednodušších variant je lineární SVM, jejíž výsledkem je čistě lineární klasifikátor[9].
Další z variant je One-Class SVM. Používá pro detekci odlehlých hodnot (outliers). Jedná se o metodu strojového učení bez učitele, která se učí rozhodovací funkci pro detekci novosti (novelty) a klasifikuje nová data jako podobná, nebo odlišná od trénovací sady [1].



Obrázek 2.2: Novelty detection[2]

[2]https://scikitlearn.org/stable/auto_examples/svm/plot_oneclass.html

3 Implementace

Prvním krokem byl výběr datasetu. Pro detekci anomálií jsem si vybral dataset *10122018-104Mega*. Pomocí nástroje *Wireshark* jsem provedl analýzu datasetu a komunikace mezi dvěma zařízeními. Dataset obsahuje 102 971 paketů. Jedná se o běžnou komunikaci pomocí protokolu *IEC 104*. Nástroj *Wireshark* umožňuje export paketů do formátu *CSV*, který jsem zvolil pro další práci s datasetem.

Pro projekt jsem zvolil programovací jazyk *Python3*. Knihovnu *sklearn* pro implementaci *One-Class SVM* a *confusion_matrix*, dále knihovnu *numpy* a *matplitlib*. Projekt se sestává ze dvou skriptů: *model.py*, který obsahuje samotnou implementaci detekce anomálií a skriptu *generator.py* sloužícího ke generování anomálních dat do formátu *csv*.

Data jsou nejdříve načtena do proměnné řádek po řádku. Ze souboru je odstraněn první řádek, který obsahuje formát dat. Následně vybírám sloupce, které obsahují relevantní data pro detekci anomálií. Jako relevantní sloupce $\langle time, source\_IP, destination\_IP, length \rangle$. Data jsou poté rozdělena na data trénovací a na data testovací v poměru 70 : 30. Trénovací i testovací data jsou dále rozdělena dle směru komunikace (Master-Slave, Slave-Master) podle *source_IP* a *destination_IP*.

Z dat je poté vytvořen krabicový graf a spočítáno pravidlo tří sigma pro oba směry komunikace na základě velikosti jednotlivých paketů, dále je ještě zobrazen graf ukazující rozdělení dat dle inter-arrival time a velikosti paketů.

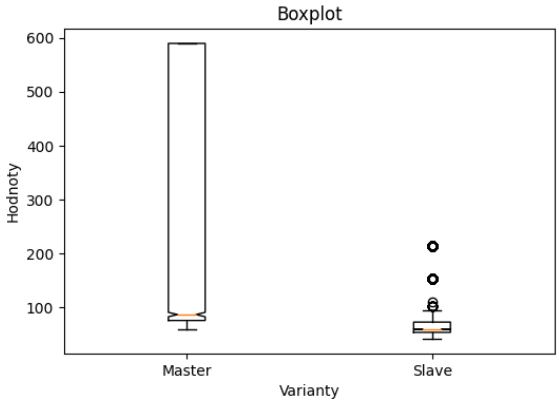
V další části jsou data dále připravována pro trénování *One-Class SVM* modelu. Jsou spočítány velikosti paketů pro oba směry komunikace, spočítána průměrná velikost paketů v daném časovém okně (určené proměnnou *interval_len*), dále spočítána celková velikost dat a počet paketů v daném časovém okně. Pro testovací data jsou provedeny stejné operace a spočítány stejné statistiky. Trénovací data jsou následně dány na vstup *One-Class SVM* modelu, který se následně vytrénuje pomocí funkce *train_svm* a uloží se. Následně je na modelu ověřena úspěšnost ve funkci *evaluate* pomocí testovacích dat, která jsou posuzována modelem. Na modelu taktéž testuji data anomální, které lze vygenerovat pomoci skriptu *generator.py*.

V další kapitole se podíváme na provedené experimenty.

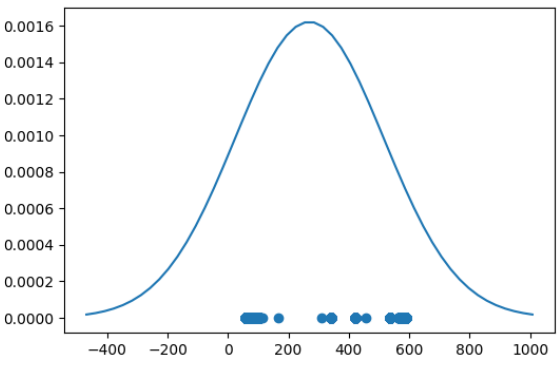
4 Experimenty

Experiment 1.

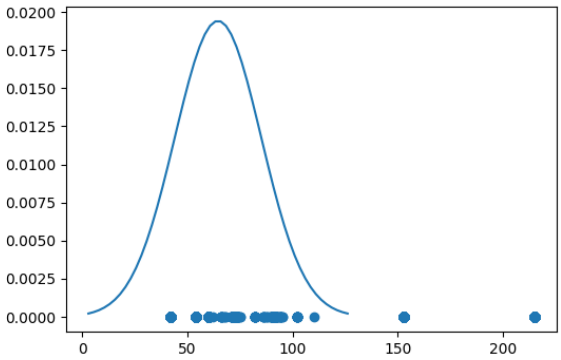
V prvním experimentu jsem si vyzkoušel implementovat techniku boxplot a pravidlo tří sigma. Na prvním obrázku je vidět srovnání obousměrné komunikace (Master → Slave, Slave → Master) na základě velikosti paketů. Na druhém a třetím obrázku je pravidlo tří sigma. Na druhém je ve směru Master → Slave a na třetím obrázku Slave → Master vypočítané opět pro velikost paketu.



Obrázek 4.1: Boxplot, vlevo M->S, vpravo S->M



Obrázek 4.2: Graf pravidla tří sigma pro Master → Slave



Obrázek 4.3: Graf pravidla tří sigma pro Slave → Master

Experiment 2.

V druhém experimentu jsem se zaměřil na jednosměrnou komunikaci, konkrétně ve směru Master → Slave. Trénoval jsem dva modely. Pro první model jsem zvolil kombinaci dvou parametrů (inter-arrival time a velikost paketu). Druhý model pracuje s časovým oknem 60s (ovlivněným proměnnou *interval_len*) a na vstupu má parametry v kombinaci celková velikost paketů a počet paketů v daném okně. Nastavení pro trénování modelu jsem u prvního modelu neměnil a ponechal základní nastavení. U druhého modelu jsem nastavení změnil na: $gamma = 0.001$ a $nu = 0.07$. Výrazný vliv na přesnost druhého modelu měla velikost časového okna. Při časovém okně 60 vteřin dosáhl model úspěšnosti 78.261% a při zmenšení časového okna na 20 sekund dosáhl model úspěšnosti 92.727%, což byl nejlepší výsledek.

Přesnost 1. modelu: 90.398%
Přesnost 2. modelu: 92.727%

Experiment 3.

Ve třetím experimentu jsem se vyzkoušel taktéž jednosměrnou komunikaci, ale oproti předchozímu experimentu to byla komunikace ze směru Slave → Master. Opět jsem trénoval dva modely. První model dal předpokládané výsledky, ale u druhého modelu se mi dostalo zajímavého zjištění, jelikož se stejnými parametry vykazoval podstatně horší výsledky než v předchozím experimentu. S časovým oknem 60 vteřin a stejnými *gamma* a *nu* byla přesnost pouze 43.478%. Zkoušel jsem kombinace *gamma* a *nu* parametrů, ale přesnost se nezlepšila. Výrazného zlepšení jsem dosáhl opět pouze pomocí časového okna. Při časovém okně 20 vteřin byla přesnost 72.363% a nejlepšího výsledku jsem dosáhl s *interval_len* 15 sekund a to 82.465%. S časovým oknem 2 minuty byla přesnost pouze 47.826%

Přesnost 1. modelu: 91.984%
Přesnost 2. modelu: 82.465%

Experiment 4.

Pro čtvrtý experiment jsem se rozhodl, že komunikaci nebudu separovat a ponechám ji v obou směrech Master → Slave i Slave → Master. Dle mého očekávání se přesnost zhoršila. Přesnost prvního modelu byla 77.025%. U druhého modelu s časovými okny šla přesnost ještě více dolů. Nejlepších výsledků jsem dosáhl s velikostí okna 20 vteřin, kdy přesnost dosahovala 59.636%.

Přesnost 1. modelu: 77.025%
Přesnost 2. modelu: 59.636%

Experiment 5.

Pátý experiment jsem obohatil o anomální komunikaci, kterou jsem pomocí skriptu *generator.py* umístil do *fake_ data.csv* souboru. Skript vygeneroval celkem 10000 paketů. Vygenerované pakety se tváří jako komunikace normální. Vždy je vygenerována dvojice paketů. Jeden paket má zdrojovou IP Mastera, cílovou Slave, inter-arrival time je náhodně vygenerována v intervalu od $\langle 0.00001, 0.5 \rangle$ sekund s tím, že nejpravděpodobnější čas je kolem 0.001s, dále typ protokolu je IEC104/TCP, poté je náhodně vygenerována velikost paketu od $\langle 50, 1000 \rangle$ bytů a následně je dodáno info paketu. Komunikace byla opět ve směru Master → Slave. Výsledná přesnost modelů byla příjemným překvapením, kdy první model dosáhl přesnosti 92.554% a druhý model dosáhl přesnosti 94.429% s časovým oknem 20 vteřin.

Přesnost 1. modelu: 92.554%
Přesnost 2. modelu: 94.429%

Confusion matrix pro 1. model

		Předpověď	
		Ano	Ne
Skuteč.	Ano	15112 (TP)	1605 (FN)
	Ne	12 (FP)	4987 (TN)

Confusion matrix pro 2. model

		Předpověď	
		Ano	Ne
Skuteč.	Ano	255 (TP)	20 (FN)
	Ne	0 (FP)	84 (TN)

Experiment 6.

V šestém experimentu jsem se pokusil o implementaci anomální komunikace ve směru Slave → Master. Skript opět na konec *csv* souboru dogeneroval 10000 paketů a mají stejnou strukturu jako v experimentu 5. Přesnost 1. modelu dosáhla 94.054% a druhého modelu 86.582%. U druhého modelu opět nejlepší hodnota časového okna byla 15 vteřin. Při velikosti okna 20 vteřin byla přesnost 78.830% a při 60 vteřinách pouze 57.024%. Zajímavým úkazem bylo, že anomální data model vždy odhalil a správně vyhodnotil při všech velikostech časového okna.

Přesnost 1. modelu: 94.054%
Přesnost 2. modelu: 86.582%

Confusion matrix pro 1. model

		Předpověď	
		Ano	Ne
Skuteč.	Ano	13037 (TP)	1136 (FN)
	Ne	4 (FP)	4995 (TN)

Confusion matrix pro 2. model

		Předpověď	
		Ano	Ne
Skuteč.	Ano	301 (TP)	64 (FN)
	Ne	0 (FP)	112 (TN)

5 Závěr

V projektu jsem prezentoval několik metod pro detekci anomálií, které jsem následně implementoval a vyhodnotil. Pravidlo tří sigma a boxplot nebyly moc vhodnými metodami pro detekci vzdálených nebo anomálních hodnot v mém případě. Proto jsem implementoval *One-class SVM*, které bylo o poznání lepší a vhodnější. Z provedených experimentů je zřejmé, že modely vytrénované na separované komunikaci dávaly lepší výsledky, než na komunikaci neseparované. Pokud by model mohl trénovat na více neanomálních datech, tak si myslím, že by přesnost mohla být ještě o něco lepší.

Projekt jako celek byl pro mě jednoznačným přínosem. Kombinuje dvě zajímavé oblasti, konkrétně počítačové sítě a strojové učení. Opět jsem se naučil něco nového a rozhodně bylo zajímavé projekt vypracovat, za což jsem rád.

Filip Weigel

Literatura

[1] *One-class SVM with non-linear kernel (RBF)*. Dostupné z: https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html.

[2] Burgetová, I., Matoušek, P. a Ryšavý, O. Anomaly Detection of ICS Communication Using Statistical Models. In: *Proceedings of the 17th International Conference on Network Service Management (CNSM 2021)*. Institute of Electrical and Electronics Engineers, 2021, s. 166–172. Dostupné z: https://www.fit.vut.cz/research/publication/12509. ISBN 978-3-903176-36-2.

[3] Matoušek, P. *Description and analysis of IEC 104 Protocol*. Faculty of Information Technology, Brno University of Technology, 2017.

[4] Wikipedia. *Anomálie — Wikipedia, The Free Encyclopedia* [http://cs.wikipedia.org/w/index.php?title=Anom%C3%A1lie&oldid=20622205]. 2022. [Online; accessed 19-April-2022].

[5] Wikipedia. *Boxplot — Wikipedia, The Free Encyclopedia* [http://cs.wikipedia.org/w/index.php?title=Boxplot&oldid=20872546]. 2022. [Online; accessed 19-April-2022].

[6] Wikipedia. *Normální rozdělení — Wikipedia, The Free Encyclopedia* [http://cs.wikipedia.org/w/index.php?title=Norm%C3%A1ln%C3%AD%20rozd%C4%9Blen%C3%AD&oldid=20872009]. 2022. [Online; accessed 19-April-2022].

[7] Wikipedia. *Počítačový útok — Wikipedia, The Free Encyclopedia* [http://cs.wikipedia.org/w/index.php?title=Po%C4%8D%C3%ADta%C4%8Dov%C3%BD%20%C3%BAtok&oldid=20811704]. 2022. [Online; accessed 19-April-2022].

[8] Wikipedia. *Pravidlo tří sigma — Wikipedia, The Free Encyclopedia* [http://cs.wikipedia.org/w/index.php?title=Pravidlo%20t%C5%99%C3%AD%20sigma&oldid=20339519]. 2022. [Online; accessed 19-April-2022].

[9] Wikipedia. *Support vector machines — Wikipedia, The Free Encyclopedia* [http://cs.wikipedia.org/w/index.php?title=Support%20vector%20machines&oldid=21150122]. 2022. [Online; accessed 19-April-2022].