VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE ANOMÁLIÍ V PRŮMYSLOVÝCH ŘÍDICÍCH SYSTÉMECH NA ZÁKLADĚ STROJOVÉHO UČENÍ

MACHINE LEARNING-BASED ANOMALY DETECTION IN INDUSTRIAL CONTROL SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Kateryna Tsymbal

AUTHOR

VEDOUCÍ PRÁCE — Ing. Ondřej Pospíšil

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Bc. Kateryna Tsymbal ***ID:*** 209239

Ročník: 2 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Detekce anomálií v průmyslových řídicích systémech na základě strojového učení

POKYNY PRO VYPRACOVÁNÍ:

Student se v práci zaměří na možnosti využití strojového učení pro detekci anomálií v průmyslových řídicích systémech. V teoretické části se zaměří na popis problematiky detekce anomálií v průmyslových řídicích systémech a provede souhrn současného stavu v této oblasti. Poté provede souhrn současného stavu veřejně dostupných datasetů pro průmyslové řídicí systémy, jednotlivé datasety kategorizuje a popíše. Dále se zaměří na detekce anomálií na základě procesních dat ze zařízení. V praktické části si student zvolí jeden hlavní dataset pro průmyslová procesní data a vytvoří model pro detekci anomálií na základě strojového učení. Student otestuje algoritmy (využije také hlubokého učení) na vybrané množině dat a okomentuje jejich výběr a vhodnost. Zaměří se také na porovnání výsledků přístupů strojového učení s učitelem a bez učitele. Provede rozbor důležitosti vstupních parametrů v návaznosti na výsledné detekce. Otestuje a okomentuje závislosti vstupních parametrů a jejich dopady na výsledky. Nakonec provede validaci svých řešení.

DOPORUČENÁ LITERATURA:

[1] MOKHTARI, Sohrab, et al. A machine learning approach for anomaly detection in industrial control systems based on measurement data. Electronics, 2021, 10.4: 407.

[2] GÉRON, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.", 2019.

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 19.5.2023

Vedoucí práce: Ing. Ondřej Pospíšil

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Hlavním cílem této diplomové práci je navržení systému detekce anomálií a narušení v průmyslových řídících systémech s pomocí strojového učení. Teoretická část práce poskytuje základní teoretický přehled o průmyslových řídicích systémech a jejich bezpečnosti. Dále jsou zmíněny poznatky o technikách detekce anomálií a možných výzvách v této oblasti. V poslední řadě byla v teoretické části provedena rešerše různých řešení detekce anomálií v průmyslových řídících systémech pomocí strojového učení. V praktické části jsou aplikovány algoritmy strojového učení na zvolenou datovou sadu HAI. Na závěr jsou shrnuty poznatky o vhodnosti užitých algoritmů a možnosti dalšího výzkumu. Účelem této diplomové práce je zvýšení bezpečnosti průmyslových řídicích systémů, a výsledky mohou sloužit jako podklad pro budoucí vývoj účinnějších metod detekce anomálií v této oblasti.

KLÍČOVÁ SLOVA

Automatizace, bezpečnost, binární klasifikace, datová sada HAI, detekce anomálií, PLC, procesní data, průmyslové řídící systémy, SCADA, strojové učení, umělá inteligence.

ABSTRACT

The main goal of this thesis is to design a system for anomaly and intrusion detection in industrial control systems using machine learning. The theoretical part of the thesis provides a basic theoretical overview of industrial control systems and their security. Furthermore, knowledge about anomaly detection techniques and potential challenges in this area are discussed. Lastly, the theoretical part has reviewed various solutions for anomaly detection in industrial control systems using machine learning. In the practical part, machine learning algorithms are applied to the selected HAI dataset. Finally, the findings on the suitability of the used algorithms and the possibilities for further research are summarized. The purpose of this thesis is to improve the security of industrial control systems, and the results can serve as a basis for the future development of more effective methods for anomaly detection in this area.

KEYWORDS

Anomaly detection, artificial intelligence, automation, binary classification, HAI dataset, industrial control systems, machine learning, PLC, process data, SCADA, security.

TSYMBAL, Kateryna. *Detekce anomálií v průmyslových řídicích systémech na základě strojového učení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 89 s. Diplomová práce. Vedoucí práce: Ing. Ondřej Pospíšil

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Kateryna Tsymbal
VUT ID autora:	209239
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Detekce anomálií v průmyslových řídicích systémech na základě strojového učení

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu diplomové práce, panu Ing. Ondřeji Pospíšilovi, za odborné vedení, užitečné konzultace, trpělivost a cenné rady při tvorbě práce.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Průmyslové řídicí systémy jsou souhrnným pojmem používaným k popisu různých typů řídicích systémů a souvisejícího přístrojového vybavení, které zahrnují zařízení, systémy, sítě a řídicí prvky používané k provozu a automatizaci průmyslových procesů. V současné době se prakticky ve všech odvětvích průmyslu používají průmyslové řídící systémy ke zdokonalení všech procesů a jejich automatizaci. S technickým pokrokem vzniká čím dál tím vyšší potřeba zrychlení a automatizace těchto procesů. Tento pokrok s sebou ovšem přináší nemalá rizika. Diplomová práce se zabývá možnostmi využití, a aplikováním strojového učení pro detekci anomálií v průmyslových řídicích systémech.

V první části práce jsou shrnuty základní informace o průmyslových řídících systémech se zaměřením na detekci anomálií pomocí strojového učení. Jsou zde probrány zejména současné výzvy v této oblasti, možné způsoby detekce anomálií, typy anomálií, a také situace vedoucí ke vzniku anomálií. Závěrem této kapitoly je shrnutí dostupných datových sad v rámci průmyslových řídících systémů, jejich nedostatky a provedení rešerše odborných článků na téma detekce anomálií, či narušení pomocí strojového učení.

V druhé kapitole jsou zahrnuty obecné informace o oblasti umělé inteligence. Je zde probráno rozdělení této oblasti a základní informace o funkčnosti strojového učení a neuronových sítí

Třetí kapitola je věnována podrobné analýze datové sady HAI ((HIL-based Augmented Industrial Control System). Jsou zde popsány procesy, ze kterých jsou data sbírána, schéma těchto procesů, a jednotlivé scénáře útoků na popsané testovací prostředí (testbed). Závěrem této kapitoly je porovnání existujících řešení v rámci datové sady HAI z hlediska detekce anomálií pomocí strojového učení.

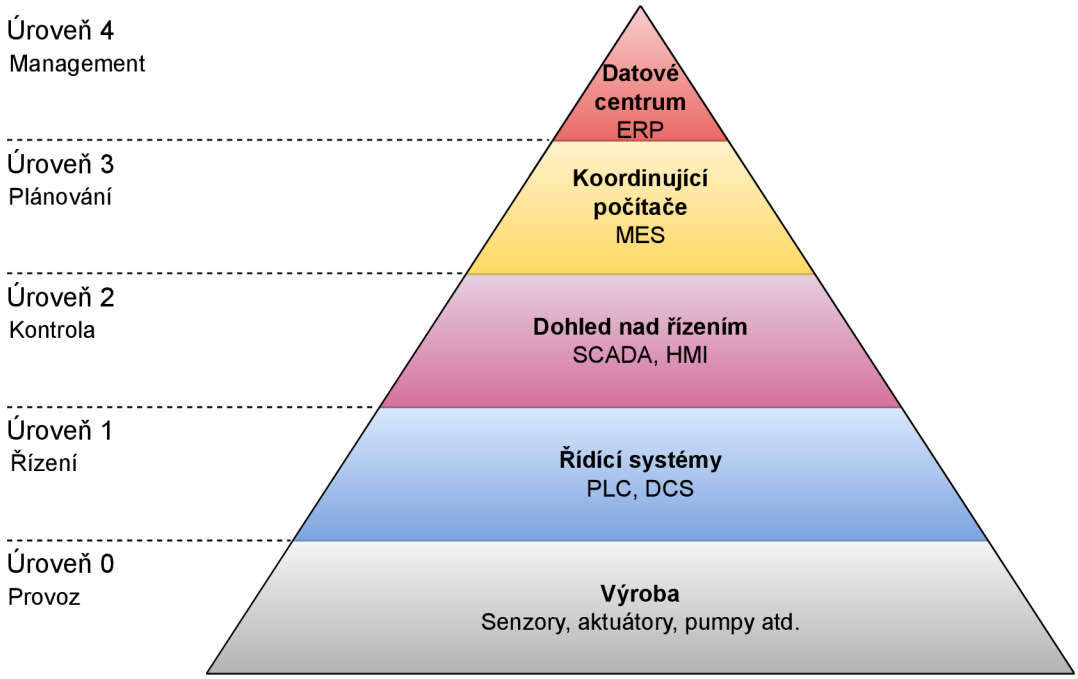
Čtvrtá kapitola zahrnuje praktickou část diplomové práce, kde jsou zpočátku shrnuty základní přístupy k učení modelů strojového učení a další důležité informace o praktickém využití algoritmů strojového učení. Následně jsou vyhotoveny a analyzovány tři základní přístupy pro detekci anomálií – učení s učitelem, neuronové sítě, a učení bez učitele. Každý z těchto přístupů je detailně popsán. Rovněž jsou analyzovány výsledky jednotlivých vytvořených modelů pro každé řešení.

V poslední, páté kapitole jsou zahrnuty výsledky práce s popisem, jak těchto výsledků bylo dosaženo a možným rozšířením této problematiky do budoucna.

1 Detekce anomálií v průmyslových řídicích systémech

Průmyslové řídicí systémy (*Industrial Control Systems* – ICS) jsou základním stavebním kamenem pro automatizaci dnešních průmyslových procesů. Součástí ICS je několik typů řídicích systémů a souvisejících komponent [1]. Tyto systémy kombinují distribuované výpočty s monitorováním a řízením fyzických procesů a jsou zodpovědné především za sběr dat, monitorování systému, automatické řízení a kontrolu průmyslových procesů, přičemž vše se odehrává v reálném čase [2].

ICS se široce používají v důležitých oblastech, jako je průmyslová výroba, inteligentní sítě, doprava, úprava vody, výroba a distribuce energie atd. Jedná se tedy o široké pole působnosti, ve kterém z velké části figurují prvky kritické infrastruktury. V rámci ICS funguje řada vzájemně propojených systémů, které řídí provoz (např. v elektrárně) a tím pádem by i malá závada mohla ohrozit výkonnost celé výroby [3]. Na obr. 1.1 lze sledovat rozdělení na pět různých úrovní referenčního modelu pro architekturu podniků [4] (tzv. automatizační pyramida – definováno v ANSI/ISA 95), která znázorňuje integrované vrstvy technologií používaných ve výrobě nebo průmyslu v kombinaci s úrovní řízení, od spodní části pyramidy, kde je realizována samotná fyzická akce, až po vrchol, kde probíhá plánování podnikových zdrojů.



Obr. 1.1: Pětivrstvá architektura ICS.

Existuje několik typů a komponentů ICS, které jsou následně probrány s ohledem na jejich umístění v rámci automatizační pyramidy [5, 6, 7, 8]:

- **Úroveň 0 (provoz)** – tato úroveň zahrnuje zařízení tvořící procesní část provozu, kde dochází k realizaci instrukcí (procesních dat), které jsou vyhodnoceny zařízeními vyšších úrovní. Tohle funguje obousměrně, tedy jedno zařízení realizuje na základě procesních dat určitou akci, a jiné zařízení akci snímá a generuje korespondující procesní data.
  - **Senzory a akční členy** – zařízení, která propojují kybernetické prostředí s fyzickým světem. Akční členy (aktuátory) jsou zařízení, která přijímají procesní data a tím manipulují s mechanickými součástmi (např. ventily, spínače, relé). Senzory hlásí údaje z výroby do řídicích jednotek, kde proběhne vyhodnocení dalšího postupu. Na základě těchto informací mohou být odeslány další instrukce do akčních členů.
- **Úroveň 1 (řízení)** – na této úrovni se řídí zařízení, která vykonávají fyzickou práci na úrovni provozu. Zařízení na této úrovní přijímají data ze všech zařízení na nižší úrovní, tyto informace vyhodnocují a rozhodují, jaký bude další postup pro dokončení naprogramované úlohy. Na této úrovni fungují programovatelné logické automaty (*Programmable Logic Controller* – PLC) a distribuované řídící systémy (*Distributed Control System* – DCS).
  - **PLC** – typ hardwaru, který se používá jako řídicí součást celého systému. Zajišťuje také místní řízení probíhajících procesů prostřednictvím senzorů a akčních členů. V DCS se PLC používají jako místní řídicí jednotky v rámci nadřazeného řídicího schématu. Do PLC je obvykle integrován PID regulátor (*Proportional Integral Derivative*), což je modul udržující proměnnou v rámci nastavených parametrů (tedy např. rozhodování, kdy zapnout/vypnout topení/chlazení pro udržení stálé teploty).
  - **DCS** – jedná se o systém, který se používá k řízení výrobních systémů nacházejících se na jednom místě. V systému DCS je do řídicí jednotky odeslána požadovaná hodnota, která je schopna dát pokyn např. ventilu aby pracoval tak, aby byla provedena požadovaná akce. Data ze systému DCS mohou být buď uložena pro budoucí použití, použita pro jednoduché řízení procesu, nebo pro pokročilé strategie řízení s daty z jiné části výroby. Každý DCS využívá centralizovanou řídicí smyčku dohledu k řízení více místních zařízení, která jsou součástí celého výrobního procesu.
- **Úroveň 2 (kontrola)** – tato úroveň je známá jako úroveň dohledu. Odehrává se zde vzdálená koordinace a monitoring zařízení nižší úrovně pomocí systémů pro dohled, řízení a sběr dat (*Supervisory Control and Data Acquisition* – SCADA) a rozhraní člověk-stroj (*Human Machine Interface* – HMI).
  - **SCADA** – systém umožňující monitorování a vzdálené řízení průmyslo-

vých a jiných technických zařízení prostřednictvím centralizovaného řídicího systému. Díky tomuto systému je možné automatizovat provádění instrukcí a shromažďování dat na dálku. Mezi hlavní činnosti tohoto systému patří řízení místních operací, jako je např. otevírání nebo zavírání ventilů, sběr dat ze senzorů, monitorování místního prostředí a hledání chybových stavů;

  - **HMI** – panel, který zprostředkovává činnost podobnou grafickému uživatelskému rozhraní (*Graphical User Interface* – GUI). HMI umožňuje interakci mezi strojem a lidskou obsluhou. Slouží k zobrazování důležitých informací a také ke konfiguraci potřebných hodnot ke správnému fungování konkrétního procesu.

- **Úroveň 3 (plánování)** – tato úroveň využívá počítačový řídicí systém známý jako výrobní informační systém (*Manufacturing Execution System* – MES). MES monitoruje celý výrobní proces v samotné továrně od surovin až po hotový výrobek. To umožňuje managementu přesně vidět, co se děje, a na základě těchto informací přijímat rozhodnutí.
- **Úroveň 4 (management)** – vrchol pyramidy tvoří úroveň správy celé infrastruktury. Tato úroveň využívá integrovaný systém řízení společnosti, který je známý jako plánování podnikových zdrojů (*Enterprise Resource Planning* – ERP). Zde má vedení společnosti přehled o svých operacích a může je řídit. Díky tomu může podnik sledovat všechny úrovně podnikání od výroby, přes prodej, nákup až po finance a mzdy a mnoho dalších. ERP je obvykle soubor různých počítačových aplikací, které mají za úkol kontrolovat a spravovat vše, co se ve společnosti děje.

Průmyslové řídicí systémy jsou ze své podstaty náročnější na zabezpečení než klasické IT systémy. Některé součásti ICS jsou v provozu neustále a jakékoliv prostředky zabezpečení mohou potenciálně omezit jejich výkon. Jakékoli odstávky vzniklé prováděním změn nebo instalací aktualizací těchto systémů musí být s dostatečným předstihem naplánovány, aby byla zajištěna naprosto minimální úroveň narušení dostupnosti služeb. Vzhledem k tomu, že organizace po celém světě v zájmu zvýšení obchodní a provozní efektivity zpřístupnily více funkcí ICS podnikovým sítím a cloudu, otevřely nové potenciální vektory útoku [9, 10, 11] na své průmyslové řídicí systémy. Útočníkům se často daří získat přístup do podnikových sítí a v mnoha případech i do prostředí ICS a způsobit různá narušení, jejichž dopady sahají od triviálních až po život ohrožující [6].

S rozvojem informačních a komunikačních technologií se zároveň zvyšuje míra a promyšlenost kybernetických útoků. V případě průmyslových řídících systémů, obzvlášť z důvodu jejich značného využití v kritické infrastruktuře, je nutné posilovat míru zabezpečení. Tohoto cíle lze dosáhnout jak pomocí detekce anomálií, tak

i s využitím detekce narušení [12]. Účelem detekce anomálií je vyhodnotit rozdíl mezi shromážděnými daty a referenčním standardem chování a tím odhalit případné narušení [13]. Anomálie mohou být způsobeny chybami v datech, ale někdy také svědčí o novém procesu, který vykazuje odchylku od klasického chování. Taková odchylka může být způsobena provozní anomálií (tedy softwarovou či hardwarovou chybou v určitém zařízení), ale také kybernetickým útokem.

Důležité je zaměřit se také na různou podobu dat, která jsou relevantní v rámci ICS. Bude následovat jejich stručný popis [1]:

- **Procesní data** – jedná se o vstupní (aktuátory), nebo výstupní (senzory) data v rámci ICS.
  - **Data ze senzorů** – surová data ze senzorů mají obvykle podobu údajů o napětí nebo elektrickém proudu, které reprezentují fyzikální veličiny, jako je teplota, tlak, poloha atd. Data jsou snímána v pravidelných intervalech a převáděna do digitálního formátu pro následné zpracování řídícími systémy (surová data ze senzorů jsou často nezpracovaná a mohou vyžadovat další interpretaci nebo škálování, než je bude moci řídicí systém použít).
  - **Instrukce aktuátorů** – pokyny pro aktuátory jsou příkazy, které jsou vysílány z řídicího systému, aby určovaly jeho požadovaný stav nebo činnost (např. otevření ventilu, spuštění motoru, pohyb robotické paže atd.). Instrukce mohou být zasílány v různých formátech, včetně binárních nebo ASCII kódů, v závislosti na typu aktuátoru a použitém komunikačním protokolu.
- **Protokolová data** – data používaná v komunikačních protokolech, jako je např. Modbus nebo S7, představují reprezentaci surových dat ze snímačů a instrukcí pro aktuátory ve standardizovaném formátu. Data jsou obvykle uspořádána do paketů, které obsahují informace o zdroji a cíli dat a také samotná data. Data jsou často zpracovávána nebo překládána komunikačním protokolem, aby bylo zajištěno jejich správné vysílání a přijímání.

1.1 Problematika detekce narušení a anomálií

Požadavky na zabezpečení v rámci ICS se výrazně liší od požadavků na tradiční informační systémy. V tradičních informačních systémech bezpečnost znamená, že neoprávněné osoby nebo organizace nemohou zveřejnit, upravit, ukrást nebo poškodit

[1] Užitá terminologie (procesní data a protokolová data) byla stanovena pro potřeby této diplomové práce.

řadu soukromých, citlivých či cenných dat [2]. V oblasti ICS je však bezpečnost chápána především jako zabránění nepříznivým dopadům selhání hardwaru, softwaru nebo systémů na bezpečnost výroby, osobní bezpečnost a bezpečnost majetku [1]. Bezpečnostní požadavky a výzvy v této oblasti by se daly shrnout následovně [1, 14]:

1. **Pomalá adaptace** – systémy ICS sice byly dlouho izolovány od internetu, ovšem z důvodu možnosti řídit velké množství systémů na dálku, namísto lokálního řízení. V současnosti existuje značná iniciativa vše (pokud možno) řešit na dálku prostřednictvím informačních systémů, jako je například cloud computing (tzv. outsourcing). Tohle řešení je efektivní z časového a tedy i finančního hlediska. V důsledku propojení s internetem byly systémy ICS vystaveny většímu riziku kybernetického útoku a zároveň velkému množství nových vektorů útoku.
2. **Zabezpečení v reálném čase** – v ICS je doba provozu každého zařízení omezena. Mírná odchylka, způsobená narušením nebo anomálií, může fyzické zařízení poškodit a vést k vážným průmyslovým haváriím;
3. **Omezené výpočetní zdroje** – senzory a aktuátory v průmyslových systémech mají omezené výpočetní zdroje, což ztěžuje podporu běhu bezpečnostních programů.
4. **Pevně daná strategie výroby** – je kritické, aby byla dodržena určitá strategie výroby. Při jejím porušení může docházet k nesrovnalostem ve výrobě, což může vést i k nehodám;
5. **Starší systémy** – v rámci ICS funguje značná část starších subsystémů, což z důvodu nepřetržitého provozu ztěžuje jejich modernizaci. Zařízení se často setkávají bezpečnostními hrozbami, což představuje velkou výzvu pro spolehlivou detekci anomálií a narušení.
6. **Náročná aktualizace** – aktualizace softwaru je z důvodu využívání staršího hardware náročná a kvůli snaze o zajištění kontinuity výroby bez odstávek často opomíjena.
7. **Špatné zabezpečení průmyslových protokolů** – se zavedením internetu se průmyslové protokoly, které byly původně v uzavřeném prostředí bezpečné, stávají v otevřeném prostředí zranitelnými vůči kybernetickým útokům. Tím pádem se zvyšuje pravděpodobnost, že důležitá a citlivá procesní data budou vystavena útočníkům.

Systémy pro odhalení průniku (*Intrusion Detection System* – IDS) v rámci ICS jsou fyzická zařízení nebo softwarové aplikace (nebo jejich kombinace), které monitorují chování ICS za účelem odhalení škodlivých aktivit nebo porušení bezpečnostních zásad sběrem a analýzou všech dostupných dat. V případě jakéhokoliv bezpečnost-

[2] Ve zkratce to znamená zajištění triády CIA (*Confidentiality, Integrity, Availability*), tedy dodržení důvěrnosti, integrity a dostupnosti důležitých dat či služeb.

ního incidentu je nutno provést určité akce. Může se jednat o snahu, aby k incidentu vůbec nedošlo, tedy preventivní opatření. V tom horším případě, když už je škoda napáchána, musí být se situací obeznámen správce systému, který realizuje jistá nápravná opatření k mitigaci škod, aby došlo k co možná nejmenšímu narušení kontinuity výroby [1].

Bude následovat výpis možných způsobů detekce anomálií v závislosti na typu dat:

- **Data ze senzorů** – detekce anomálií dat produkovaných senzory se používá k detekci anomálií v údajích ze senzorů, které mohou indikovat poruchu nebo abnormální chování sledovaného systému. Například náhlý pokles teploty nebo náhlý nárůst tlaku může indikovat anomálii ve sledované části procesu.
- **Instrukce aktuátorů** – tento způsob detekce se zaměřuje na sledování příkazů zasílaných zařízením, která řídí provoz ICS. Tento přístup se používá k detekci anomálií v odesílaných příkazech, které mohou indikovat kybernetický útok nebo jiné abnormální chování. Útočník se například může pokusit odeslat příkazy, které způsobí, že systém bude fungovat nežádoucím či nebezpečným způsobem.
- **Protokolová data** – tento přístup se používá k detekci anomálií v přenosu zpráv, které mohou indikovat kybernetický útok nebo jiné abnormální chování. Například abnormální vzorce komunikace, jako je neobvykle vysoký počet paketů odeslaných do určitého zařízení, mohou indikovat útok.

Detekce anomálií v systémech ICS obvykle zahrnuje kombinaci těchto přístupů k monitorování různých zdrojů dat a odhalování anomálií, které mohou naznačovat abnormální chování nebo kybernetické útoky. Volba přístupu závisí na konkrétních analyzovaných datech a typu zjišťované anomálie. Pro zajištění bezpečnosti a spolehlivosti ICS je důležité zvolit vhodný přístup a průběžně systémy monitorovat. Je tedy vhodné, v rámci zajištění spolehlivého provozu, věnovat pozornost všem neobvyklým změnám, jako jsou např. náhlé výkyvy v přenosu dat a nesrovnalosti naměřených hodnot nebo odesílaných instrukcí. Tyto abnormality lze detekovat právě s pomocí strojového učení.

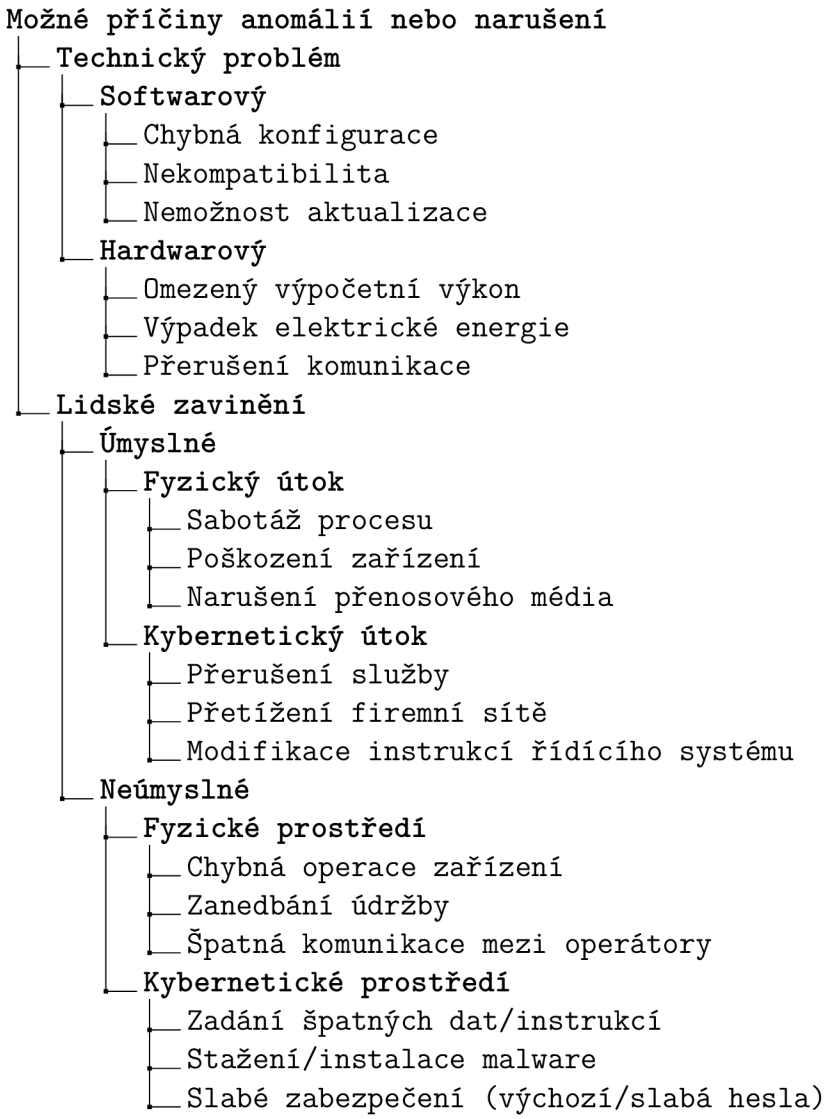
Typy anomálií v průmyslu

Rozeznání různých typů anomálií je důležité pro navržení účinného řešení pro jejich detekci. Obecně se anomálie dají rozdělit na tři kategorie [15]: *bodové*, *kontextové* a *kolektivní*. V rámci bodových anomálií dochází k odchylce jednoho nebo více vzorků dat od zbytku (referenční data) bez souvislosti s běžným vzorcem dat. Oproti tomu, pokud jednotlivé vzorky dat vypadají normálně, ale s přihlédnutím k celé množině (posloupnosti) dat dochází k odchylkám, dá se konstatovat, že se jedná o kontextovou

anomálii. Posledním typem jsou kolektivní anomálie, které se vyznačují výraznou odchylkou určité skupiny dat od celé množiny. Tyto tři přístupy k detekci anomálií se využívají ve velké míře v kybernetické bezpečnosti, ale také i v oblastech, jako např. detekce podvodů, monitorování sociálních médií, provoz strojů v průmyslu a medicína [16].

Bude následovat výpis možných situací (detailněji v obr. 1.2) vedoucích ke vzniku anomálií, které mohou nastat v rámci průmyslových řídících systémů [17, 18, 19]:

- **Technické problémy** – příčina anomálie může pocházet z vnějšího vlivu (např. výpadek dodávky elektrické energie, narušení komunikace se sítí ICS atd.), ale může se také jednat o poruchu určitého zařízení (např. selhání teplotního senzoru, závada na měření průtoku vody atd.). Může se ovšem jednat o závažný problém pro konkrétní proces a je tedy nutno přijmout vhodná opatření pro napravení chyby.
- **Lidské zavinění** – v tomto případě může být mnoho možností vzniku anomálie, ať už neúmyslně, nebo úmyslně. Například se může jednat o zadání špatných hodnot z nedbalosti, ale také záměrné narušení určitého procesu. Další častou chybou je nedostatečné proškolení personálu a tím pádem dochází k určitému selhání, zejména v důsledku nedostatečných schopností určité osoby. Do této kategorie také patří kybernetické útoky. nejčastějším způsobem pro začátek kybernetického útoku je sociální inženýrství (např. podvodné e-maily pro získání administrátorského přístupu do vnitřní sítě organizace). Jakmile se útočníkovi povede získat kontrolu nad sítí a tedy i nad systémy ICS, může dle možností provést útok narušení služeb (*Denial of Service* – DoS) pro vyřazení nějaké části procesu z provozu nebo může změnit určité parametry konkrétního zařízení (ať už procesního, nebo řídícího) a tím způsobit jeho poškození.



Obr. 1.2: Možné situace vedoucí ke vzniku anomálií nebo narušení [20, 21, 22]

Možné metody detekce

Obecně existují dva základní způsoby detekce narušení: detekce založená na signaturách nebo na porovnávání vzorů, která se také označuje jako detekce založená na znalostech (*knowledge based*) a detekce založená na rozeznávání normálního a podezřelého chování, tedy detekce založená na zjišťování anomálií (*anomaly based*) [23].

Detekce založené na znalostech fungují na základě vzorů již rozpoznaných útoků. Systémy IDS monitorují síť porovnávají signatury dat, aby identifikovaly veškeré škodlivé aktivity a v případě zjištění takové aktivity vydaly varování o detekovaném incidentu. Tyto systémy mají to omezení, že pokud není signatura útoku dostupná v jejich databázi, aktivitu vyhodnotí jako nezávadnou. Přestože je většina těchto systémů opatřena mechanismy pravidelné aktualizace databáze na signatury nových

útoků, postrádají možnost detekce nových hrozeb, které ještě nejsou v databázi [24].

Systém založený na anomáliích využívá běžný profil systému nebo uživatele k určení svého rozhodovacího procesu. Tento typ systému je vhodný pro detekci útoků, které nejsou známé a nejsou zapsané v databázi signatur. Stále s sebou však přináší problémy. Jakmile neexistuje žádná databáze známých signatur anomálií či narušení, má to za výsledek vysokou míru falešně pozitivních (*false positive*) výsledků, tedy falešných poplachů. Navíc, pokud je těchto falešných poplachů velké množství, nestíhají je bezpečnostní pracovníci všechny procházet a posuzovat jejich validitu. V tomto případě lze využít technik strojového učení, díky kterým je možné procházet velké soubory dat a hledat v nich odchylky a tím zefektivnit a zrychlit proces detekce anomálií, a to nejen s pomocí dat z minulosti, ale i v reálném čase [23, 25].

1.2 Současný stav oblasti

Nejznámějším útokem na ICS v historii je malware Stuxnet. Tento malware byl vytvořen za účelem napadení systémů SCADA a jeho úkolem bylo přeprogramování PLC takovým způsobem, že nebylo možné zjistit narušení. Stuxnet je schopen upravit hodnoty, které PLC posílá procesním zařízením, přičemž uživatelům vrací hodnoty bez známek podezřelého chování. Jedná se o první známý malware, který byl vytvořen k narušení systémů ICS. Je také považován za první kybernetickou zbraň [26].

V souvislosti s Průmyslem 4.0 se začal užívat pojem průmyslový internet věcí (*Industrial Internet of Things* – IIoT). IIoT se skládá z mnoha zařízení propojených komunikačním softwarem. Výsledné systémy mohou monitorovat jednotlivé procesy, shromažďovat důležitá data, analyzovat je a na základě těchto informací jednat a inteligentně měnit své chování s minimálním zásahem člověka. Hlavním cílem IIoT je spojení všech zařízení pro řízení procesů v průmyslu přes cloud [27].

V rámci IIoT dochází k produkování velkého množství průmyslových dat, což vyžaduje výkonné výpočetní zdroje. Díky cloud computingu mohou podniky přesunout výpočetní úlohy do cloudu namísto vlastních fyzických strojů, což s sebou přináší značné výhody z pohledu financí, ovšem ale i nevýhody z pohledu bezpečnosti. V současnosti je ICS důležitou součástí průmyslového odvětví a otázky jeho bezpečnosti se stávají kritickými pro rozvoj IIoT [28].

Dle [29] jen v rámci České republiky existuje více než 1 600 průmyslových řídících systémů, které jsou dostupné z internetu. Důvod tohoto napojení na internetu může být komunikace managementu určité firmy s výrobou kvůli usnadnění plánování výroby a také kvůli samotným řídícím systémům, které spolu potřebují stále komunikovat. Může se jednat o snímací zařízení, které dříve bylo dostupné pouze lokálně, ale v současnosti může být propojeno i s webovým serverem, aby se daly hodnoty

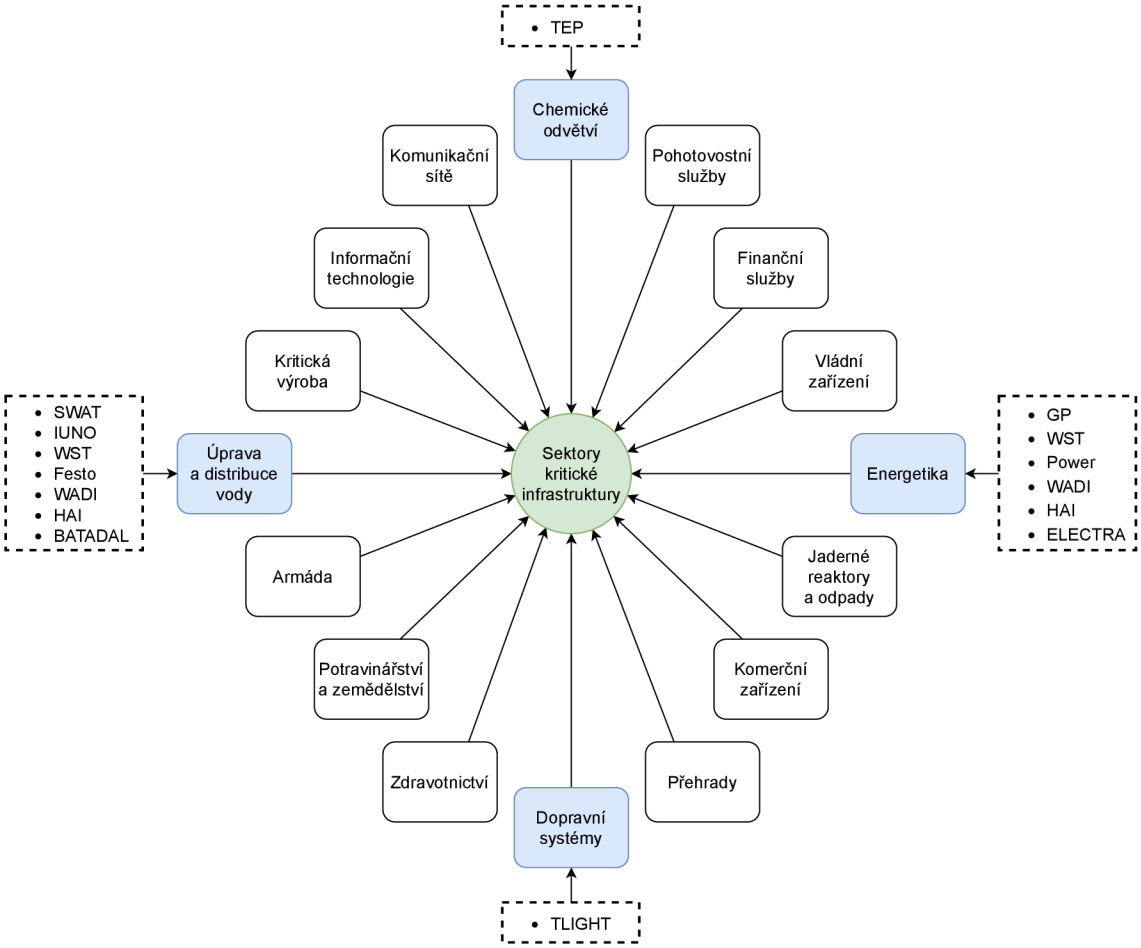
snímat na dálku nebo toto zařízení kalibrovat. Bez takového propojení již dnes většinou nedokáží velké organizace fungovat, aby zajistily spolehlivost a optimální dobu trvání výroby, či určitého procesu.

Hlavním problémem spojeným s protokoly [3], které se v ICS používají ke komunikaci je, že mnoho v současnosti používaných protokolů postrádá implementaci autentizace a šifrování komunikace a má pouze slabou nebo žádnou ochranu integrity dat. Z důvodu těchto nedostatků jsou ICS vůči průniku nebo malwaru ať už z hlediska interních, či externích hrozeb. Se znalostí funkčnosti daných protokolů a provedení průzkumu sítě je útočník schopen napáchat značné škody a vyřadit tak průmyslová zařízení z provozu, nebo je překonfigurovat a upravit tím jejich činnost [30]. Komplikací stále zůstává opomíjení těchto problémů, protože společnosti žijí v domnění, že preventivní opatření nejsou třeba. Jakmile už dojde k bezpečnostnímu incidentu zaviněného kybernetickým útokem, musí se aplikovat nápravná opatření, která jsou nákladnější a obnovení výroby či procesu může trvat týdny až měsíce.

[3] Jedná se o protokoly běžně používané v systémech ICS, např. Modbus, PROFINET, DNP3, S7COMM atd.

1.3 Dostupné datové sady pro průmysl

Ve [30] je zpracována rešerše nejznámějších datových sad z oblasti průmyslu. Datové sady pro průmysl jsou v současnosti omezeny jen na pár odvětví kritické infrastruktury (zejména úprava vody, chemie, plynovody v energetice a dopravní systém). Je z tohoto hlediska tedy opomenuta řada důležitých odvětví kritické infrastruktury a především odvětví průmyslu, která nespadají do kritické infrastruktury. Podle Agentury pro kybernetickou bezpečnost a bezpečnost infrastruktury (*Cybersecurity and Infrastructure Security Agency* – CISA) existuje 16 sektorů kritické infrastruktury[4]. Na obrázku 1.3 lze vidět jednotlivé sektory kritické infrastruktury podle CISA a k nim náležící dostupné datové sady. Navíc datové sady obsahují pouze určité typy útoků, což omezuje možnosti z hlediska detekce anomálií a narušení.



Obr. 1.3: Sektory kritické infrastruktury dle CISA a datové sady [30]

[4]Dle Asociace kritické infrastruktury České republiky (AKI ČR) se rozlišuje 9 sektorů kritické infrastruktury, a to energetika, vodní hospodářství, potravinářství a zemědělství, zdravotnictví, doprava, komunikační a informační systémy, finanční trh a měna, nouzové služby a veřejná správa [31].

Tyto datové sady lze použít k vývoji a vyhodnocení algoritmů strojového učení pro detekci a zmírnění kybernetických a fyzických útoků v průmyslových řídicích systémech. Je však důležité poznamenat, že některé z těchto datových sad obsahují uměle generované útoky, které nemusí přesně odrážet skutečné chování reálných systémů, a při interpretaci výsledků experimentů strojového učení na těchto datových sadách je tedy třeba brát tento nedostatek v potaz. Bude následovat stručný popis datových sad z obr. 1.3 [30]:

- **Secure Water Treatment** (*SWaT*) – tato datová sada obsahuje 11 dní běžného provozu a data z kybernetických a fyzických útoků ze zmenšeného reálného testovacího prostředí (testbed) průmyslové čistírny odpadních vod. Datová sada obsahuje 36 útoků provedených za poslední 4 dny z celého měření, které trvaly od několika minut až do jedné hodiny.
- **Gas Pipeline** (*GP*) – obsahuje 274 627 případů síťové komunikace mezi RTU a MTU prostřednictvím protokolu Modbus RTU. V datové sadě je zahrnuto 35 kybernetických útoků, v náhodných intervalech, sestávajících z průzkumu sítě, FDI (*respond injection, command injection*) a DoS útoků.
- **IT-Sicherheit für Unternehmensnetze Ostbayern – IT Security for Corporate Networks in Eastern Bavaria** (*IUNO*) – provoz generovaný pomocí modelu Festo Didactic představujícího prostředí vodního čerpadla, vyprazdňování a plnění ve vodní nádrži. Byly vytvořeny tři datové sady, kde každá datová sada obsahuje specifický přístup útoku typu *false data injection.*
- **BATtle of the Attack Detection Algorithms** (*BATADAL*) – tři různé simulované datové sady založené na fiktivním vodovodním systému města C, vytvořené pro soutěž v detekci kybernetických útoků. Datové sady zahrnují dvě trénovací datové sady a testovací datovou sadu. Testovací datová sada se skládá ze 407 hodinových záznamů s dalšími sedmi typy útoků.
- **Water Storage Tank and Gas Pipeline SCADA systems** (*WST*) – shromážděná ze systémů SCADA v laboratorním prostředí na Státní univerzitě v Mississippi. Obě datové sady obsahují normální data a čtyři typy útoků (dva typy útoků *false data injection*, útok typu DoS a útok průzkumem).
- **Power System Attack** (*Power*) – tři datové sady vytvořené Státní univerzitou v Mississippi a Národní laboratoří Oak Ridge. Útočné události představují útoky typu false data injection, zahrnující *remote command injection* a *relay setting change attacks.*
- **Water Distribution Testbed** (*WADI*) – data shromážděná ze zmenšené vodovodní distribuční sítě městě. Datová sada byla shromažďována po dobu 9 dnů, během nichž bylo na systém provedeno celkem 19 různých typů kybernetických útoků, včetně útoků *SQL injection*, DoS útoků a *replay* útoků.
- **Festo MPA Process Control Rig** (*Festo*) – soubor datových sad, které byly

vytvořeny pomocí zařízení pro řízení procesů úpravy vody vyvinutého společností Festo Didactic. Zařízení se skládá z nádrže na vodu, čerpadla a ventilu a bylo použito k simulaci procesu úpravy vody. Datová sada byla generována tak, že do systému byly vpraveny útoky typu *false data injection*, které vedly ke změnám hladiny vody v nádrži.

- **Tennessee Eastman Process** (*TEP*) – simulace skutečného průmyslového procesu v chemickém průmyslu. Výzkumníci znovu vytvořili nové datové sady, které obsahují více příkladů pro tréninková i testovací data. Datová sada se skládá z měření senzorů shromažďovaných v jednominutových intervalech po dobu několika měsíců. Datová sada TEP se běžně používá jako referenční soubor pro metody monitorování procesů a detekce poruch. V posledních letech výzkumníci znovu vytvořili nové datové sady s větším počtem příkladů pro trénovací i testovací data a jako způsob simulace kybernetických útoků v datové sadě TEP byly použity útoky typu *false data injection*.
- **Traffic Light Control System** (*TLIGHT*) – dvě datové sady obsahující sedm typů běžných operací, které způsobily odchylky v časovačích a výstupních hodnotách systému řízení semaforů. Útočná data byla vytvořena změnou některých hodnot.
- **HIL-based Augmented ICS Security** (*HAI*) – detailní popis v kapitole 3.1.

V návaznosti na zmíněné datové sady bude následovat shrnutí výzev, kterým čelí vývoj přístupů založených na strojovém učení. Tyto výzvy ztěžují vývoj a vyhodnocování účinných přístupů založených na strojovém učení pro detekci a zmírňování kybernetických útoků na ICS [30]:

- **Omezené scénáře útoku pro hodnocení** – vysoce cílené a specifické útoky na ICS nejsou běžné, což omezuje rozmanitost kybernetických útoků, které jsou k dispozici pro testování a hodnocení např. algoritmů strojového učení. Naproti tomu kybernetické útoky na běžné IT infrastruktury mají obvykle větší a rozmanitější vzorky, než právě zmíněné datové sady.
- **Omezený počet kvalitních a realistických datových sad** – dostupné datové sady používané pro trénování, testování a vyhodnocování přístupů založených na strojovém učení v ICS jsou zastaralé, nerealistické a mohou odrážet pouze specifické kybernetické útoky. Byly zavedeny novější datové sady, které však mohou zachycovat data ze specifických komponent nebo protokolů v prostředí ICS, což omezuje typy kybernetických útoků, které jsou k dispozici pro detekci.
- **Riziko sofistikovaných útoků** – cílem takových útoků může být zmást modely strojového do míry, aby provedly nesprávnou klasifikaci, a mohou zneužít trénovací data a předem natrénované modely, aby se vyhnuly detekci. Ačkoli byla navržena řešení a návrhy na řešení tohoto problému, není známo, zda jsou

současné přístupy využívající strojového učení odolné vůči útokům a zda jsou schopny účinně odhalit všechny typy skutečných kybernetických útoků v ICS.

- **Nedostatek metrik výkonnosti jednotlivých přístupů** – kombinace výše uvedených problémů vede k jedné z největších výzev při vývoji přístupů založených na strojovém učení, kterou je vyhodnocování reálných útoků. Neexistuje žádný standardizovaný soubor výkonnostních metrik pro měření těchto přístupů, což ztěžuje průmyslu zavádění těchto přístupů do svých systémů, zejména v kritické infrastruktuře.

Důležitým krokem ve vývoji modelů strojového učení pro zlepšení kybernetické bezpečnosti v oblasti ICS je překonání těchto výzev. Jedním ze způsobů by mohl být vývoj nových, rozmanitějších datových sad, které zachycují realistické scénáře a útoky a které lze použít pro trénování, testování a vyhodnocování modelů strojového učení. Dalším způsobem je implementace pokročilejších technik strojového učení, které jsou více odolné vůči sofistikovanosti útoků a mohou účinně odhalovat nové a neznámé kybernetické útoky. Kromě toho může vývoj standardizovaných výkonnostních metrik pomoci měřit účinnost různých přístupů založených na strojovém učení a usnadnit tak implementaci různých přístupů v praxi.

V rámci této kapitoly byla provedena rešerše na existující řešení z hlediska detekce anomálií, či narušení v rámci průmyslových řídících systémů. V rámci rešerše byly zkoumány odborné články na danou tématiku. Po provedení rešerše byla vytvořena tabulka 1.1. Parametry pro rešerši byly následující:

- **Název** – název článku konkrétního řešení.
- **Rok vydání** – rok, ve kterém byl konkrétní odborný článek vydán, přičemž byly brány v potaz řešení od roku 2017.
- **Zdroj dat** – informace o charakteristice dat, tzn. zda se jednalo o procesní, či protokolová data.
- **Průmyslová oblast** – oblast průmyslu, na kterou se článek zaměřuje.
- **Prostředí pro sběr dat** – informace o způsobu sbírání dat, tedy zda se jednalo o data z reálného prostředí, či ze simulace (např. testbed).
- **Reference** – odkaz na konkrétní článek.

Tab. 1.1: Odborné články na téma detekce anomálií, či narušení pomocí strojového učení v rámci ICS

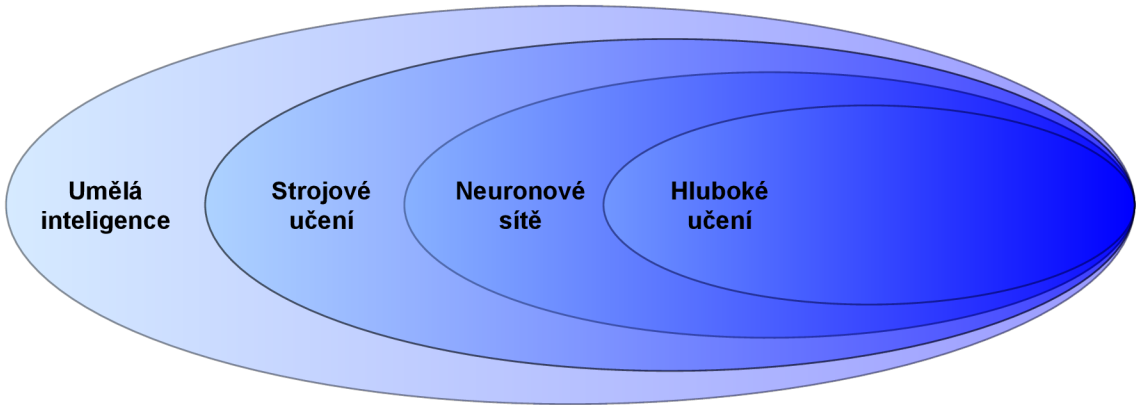
Název	Rok vydání	Zdroj dat	Průmyslová oblast	Prostředí pro sběr dat	Reference
Anomaly detection for ICS based on deep learning: a use case for aeronautical radar data	2022	protokolová data	letectví	reálné prostředí	[32]
Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM networks	2017	protokolová data	plynovod	testbed (fyzický)	[33]
Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks	2018	procesní, protokolová	úprava vody	testbed (fyzický)	[34]
Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network	2020	procesní, protokolová	úprava vody	testbed (fyzický)	[35]
Adaptive anomaly detection system based on machine learning algorithms in an industrial control environment	2021	procesní, protokolová	úprava vody, plynovod	testbed (fyzický), reálný proces	[36]
Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning	2017	procesní, protokolová	úprava vody	testbed (fyzický)	[37]
Identification of malicious activities in industrial internet of things based on deep learning models	2018	protokolová data	neuvedeno	neuvedeno	[38]
An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System	2020	procesní, protokolová	úprava vody, plynovod	testbed (fyzický), reálný proces	[39]
Anomaly detection in Industrial Control Systems using Logical Analysis of Data	2020	procesní, protokolová	úprava vody	testbed (fyzický)	[40]
Anomaly Detection of Industrial Control Systems Based on Transfer Learning	2021	procesní data	plynovod	testbed (fyzický)	[41]
Anomaly Detection for Water Treatment System based on Neural Network with Automatic Architecture Optimization	2018	procesní, protokolová	úprava vody	testbed (fyzický)	[42]
MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks	2019	procesní, protokolová	úprava vody	testbed (fyzický)	[43]
Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks	2019	procesní, protokolová	úprava vody	testbed (fyzický)	[44]
Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA	2019	procesní, protokolová	úprava a distribuce vody	testbed (fyzický)	[45]
Unsupervised Anomaly Detection for Network Data Streams in Industrial Control Systems	2020	procesní, protokolová	úprava vody	testbed (fyzický)	[46]
High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks	2018	procesní, protokolová	úprava vody	testbed (fyzický)	[47]
A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems	2020	procesní, protokolová	úprava vody	testbed (fyzický)	[48]
Misuse Intrusion Detection Using Machine Learning for Gas Pipeline SCADA Networks	2019	procesní, protokolová	úprava vody	testbed (fyzický)	[49]
New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning	2019	protokolová data	neuvedeno	neuvedeno	[50]
On the Generation of Anomaly Detection Datasets in Industrial Control Systems	2019	procesní, protokolová	elektrická energie, plyn, voda, chemie	testbed (fyzický)	[51]

2 Oblast umělé inteligence

Umělá inteligence (*Artificial Intelligence* – AI) je široký pojem pojem používaný pro klasifikaci strojů, které napodobují lidskou inteligenci. AI se používá k předvídání, automatizaci a optimalizaci úkolů, které lidé automatiky realizují každý den běžného života, jako je rozpoznávání řeči a obličeje, rozhodování, překládání a mnoho dalších [52]. Oblast umělé inteligence zahrnuje několik podoblastí:

- **Strojové učení** (*Machine Learning* – ML).
- **Neuronové sítě** (*Neural Networks* – NN).
- **Hluboké učení** (*Deep Learning* – DL).

Oblasti AI jsou znázorněny na obr. 2.1.



Obr. 2.1: Oblasti umělé inteligence [52]

2.1 Strojové učení

Strojové učení je obor, který dává počítačům schopnost učit se (podobně jako lidský mozek), získáváním znalostí na základě vstupů, jimiž mohou být jakákoliv data (např. množina číselných dat, grafy, obrázky atd.), která slouží k trénování konkrétního algoritmu. Pokud mezi sebou mají vstupní data nějaké souvislosti, model strojového učení je dokáže rozeznat a určit vyhodnocený výstup, který se následně porovná s očekávaným výsledkem pro výpočet úspěšnosti modelu. Strojové učení se často užívá k úloze klasifikace – tedy určení jisté třídy dat v závislosti na konkrétním problému. Nejčastěji se využívá binární klasifikace v rámci které se může jednat o cokoliv od rozeznávání fotek koček a psů až po detekci anomálií v průmyslových řídících systémech. Hlavním cílem ML je umožnit počítačům učit se samostatně s minimálním lidským zásahem a podle toho upravovat svá rozhodnutí [53, 54].

2.2 Neuronové sítě

Neuronová síť napodobuje fungování neuronů v lidském mozku. Jedná se o řadu algoritmů, které se snaží rozpoznat základní vztahy v souboru dat. Neuronové sítě se dokáží přizpůsobit měnícím se vstupům k nejlepšího možného výsledku. Koncept neuronových sítí, který má své kořeny v umělé inteligenci a strojovém učení, se v dnešní době využívá ve velkém množství různých oblastí [52, 55], jako je např. zdravotnictví, sociální média, bankovnictví, marketing, kybernetická bezpečnost atd.

Po určení vstupní vrstvy neuronové sítě se každému neuronu přiřadí váhy. Tyto váhy pomáhají určit důležitost dané proměnné, přičemž větší váhy přispívají k výstupu významněji než ostatní vstupy. Na samém začátků procesu fungování neuronové sítě se všechny vstupy se pak vynásobí příslušnými váhami a poté se sečtou. Poté jsou vyhodnocená data předána aktivační funkci, která určuje výstup. Pokud tento výstup překročí danou mezní hodnotu (*threshold*), dojde k neuronu a na základě toho k předání dat další vrstvě sítě. Výsledkem je, že výstup jednoho uzlu se stane vstupem dalšího uzlu. Tento proces předávání dat z jedné vrstvy do další bez tvoření smyček definuje neuronovou síť jako tzv. dopřednou (*feed-forward*) neuronovou síť [56, 57].

3 Databáze a metodologie

V této části práce je popsána zvolená datová sada pro následné testování modelu strojového učení. Navíc je zde rozebráno porovnání podobných datových sad z oblasti průmyslu s cílem nalezení optimální datové sady pro naučení vybraného algoritmu strojového učení. Dílčím cílem této práce je navíc stručný popis současných datových sad a jejich kvality z hlediska použitelnosti na detekci anomálií. Hlavní zaměření je na datovou sadu HAI (*HIL-based Augmented ICS*) Security Dataset [58]. Datová sada HAI byla vybrána konkrétně z důvodu kvalitního obsahu dat, rozsáhlého zpracování dokumentace a provedení různých scénářů útoku. Zároveň je datová sada autory pravidelně aktualizována.

3.1 Popis zvolené datové sady

Zvolená datová sada HAI byla shromážděna z realistického testovacího prostředí průmyslového řídícího systému, který byl rozšířen o simulátor HIL[1] (*Hardware-In-the-Loop*) emulující výrobu elektrické energie pomocí parní turbíny a přečerpávací vodní elektrárny. Jedná se tedy o fyzický testbed, rozšířený o virtuální prostředí HIL.

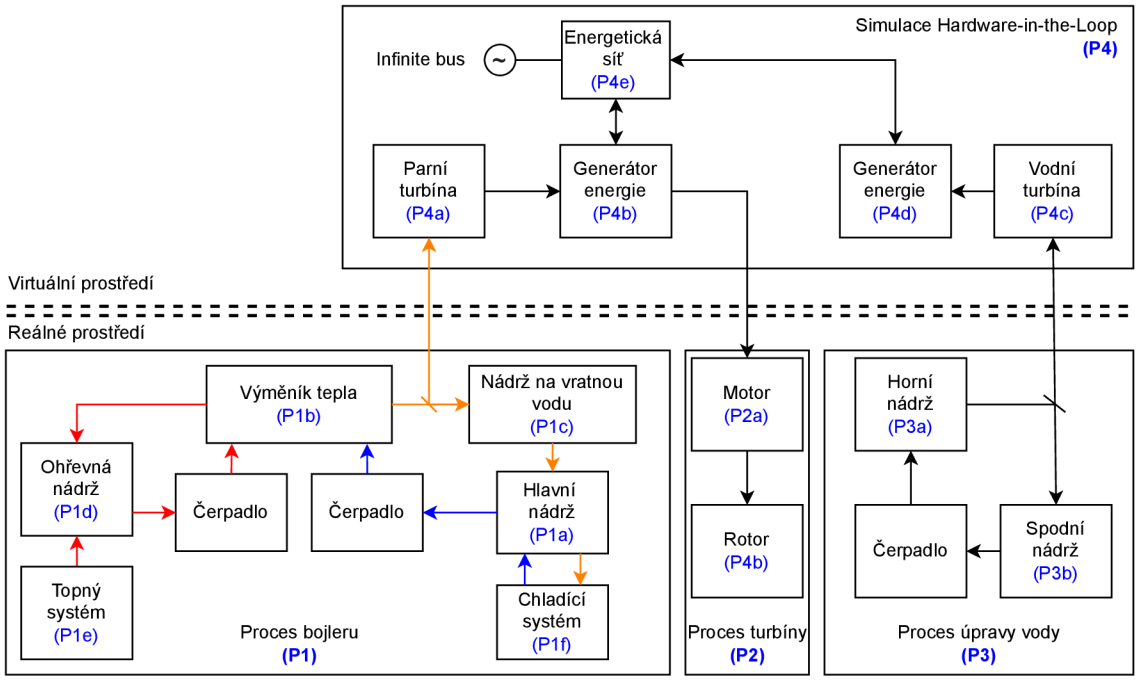
V rámci této datové sady jsou zahrnuty situace s normálním i abnormálním chováním testovacího systému ICS. HAI byl vyvinut pro účel zkoumání detekce anomálií s pomocí strojového učení. Abnormální data byla shromážděna na základě různých scénářů útoku se šesti řídícími smyčkami ve třech průmyslových řídících zařízeních: Emerson Ovation, GE Mark-VIe a Siemens S7–1500. Simulátor HIL byl implementován pomocí systému dSPACE® SCALEXIO. Simulační prostředí se tedy skládá ze čtyř hlavních zařízení a čtyř k nim náležících procesů [58] (schéma testovacího prostředí je znázorněno v obrázku 3.1):

- **Bojler (P1)** – proces bojleru zahrnuje přenos tepla mezi teplou a studenou vodou založený na nízkém tlaku a průměrné teplotě. Tímto procesem jsou řízeny tlak, teplota a hladina vody v bojleru. Rychlost otevírání a zavírání hlavního ventilu je řízena v závislosti na rychlosti otevírání parního ventilu tepelné elektrárny v simulátoru HIL a jsou sem v reálném čase přenášeny tlak a teplota hlavního potrubí a hladina vody, aby se určilo množství vyráběného výkonu. Pro řízení tohoto procesu bylo využito Emerson Ovation DCS;
- **Turbína (P2)** – tento proces zahrnuje turbínu, jejíž otáčky jsou synchronizovány s otáčkami modelu parního generátoru elektrické energie v simulátoru HIL. Navíc turbína zahrnuje otáčkoměr a čtyři sondy monitorující vibrace pro

[1] Hardware-In-the-Loop je technika využívaná při vývoji a testování reálných hardwarových zařízení (embedded systémů) ve virtuálním modelu pro vytvoření uzavřené řídící smyčky [59].

udržení konstantních otáček motoru a rozhraní HMI pro ruční upravení otáček obsluhou. Pro řízení tohoto procesu bylo využito GE Mark VIe DCS;

- **Zařízení na úpravu vody (P3)** – proces úpravy vody zahrnuje čerpání a vypouštění vody mezi horní a dolní nádrží pomocí modelu vodní turbíny v simulátoru HIL. V rámci tohoto procesu je implementováno sedm snímačů, jeden pohon a odtokový regulační ventil pro řízení průtoku a tlaku ze zpětné nádrže do hlavní nádrže a také pro řízení hladiny vody v hlavní nádrži. Hydraulický tlak, průtok a hladina vody v horní vodní nádrži jsou v reálném čase přenášeny do simulátoru HIL, aby se určilo množství vyrobené energie.
- **Simulátor HIL (P4)** – úkolem HIL simulátoru je kombinace třech řídících systémů předchozích procesů pro vytvoření systému pro výrobu elektrické energie. Skládá se ze dvou modelů synchronních generátorů (tj. generátoru s parní turbínou a generátoru přečerpávací vodní elektrárny) a jednoho modelu elektrické sítě, který zahrnuje místní poptávku po zátěži a byl připojen ke sběrnici infinite bus.



Obr. 3.1: Schéma jednotlivých procesů v simulačním prostředí [58]

Detailní popis procesů

V rámci **P1** je čerpána chladná voda v hlavní nádrži (*P1a*) do výměníku tepla (*P1b*) prostřednictvím čerpadla, které následně dodává vodu o konstantní teplotě a tlaku do nádrže na vratnou vodu (*P1c*). Topný systém (*P1e*) předává tepelnou energii do *P1b* prostřednictvím nádrže na ohřívající se vodu (*P1d*). Hodnoty teploty a tlaku vody se poté převedou na aktuální hodnoty teploty a tlaku páry do parní turbíny simulátoru HIL (*P4a*), která pohání generátor elektrické energie (*P4b*). Voda posléze proudí z *P1c* do *P1a* s konstantním průtokem, čímž se v *P1c* udržuje stálá hladina vody. Voda cirkulující v *P1a* musí být dodatečně ochlazena, a proto z ní chladicí systém (*P1f*) odebírá tepelnou energii. Energie z *P4b* je následně převedena do motoru (*P2a*), který pohání rotor (*P2b*) v **P2**. V **P3** dochází k cirkulaci vody mezi horní (*P3a*) a spodní (*P3b*) nádrží na vodu s využitím modelu vodní turbíny (*P4c*), která pohání generátor elektrické energie (*P4d*). Nakonec je v rámci **P4** výstupní energie z generátorů *P4b* a *P4d* je následně převedena do simulovaného modelu energetické sítě (*P4e*).

Detailní popis zvolené datové sady

V tab. A.1 a A.2 jsou vypsány jednotlivé datové body v rámci zvolené datové sady, tedy HAI 22.04 a jejich popis. HAI 22.04 obsahuje šest souborů CSV v rámci trénovací datové sady a čtyři soubory CSV v rámci testovací datové sady. První sloupec každého CSV souboru představuje pozorovaný čas ve formátu "RRRR-MM-DD-HH:MM:SS", zatímco dalších 87 sloupců poskytuje hodnoty zaznamenané datovými body SCADA. Poslední čtyři sloupce znační v konkrétním čase přítomnost/nepřítomnost útoku pomocí datového typu Boolean (0 – normální provoz; 1 – probíhá útok). Z těchto sloupců se první sloupec vztahuje na všechny procesy a ostatní tři sloupce se vztahují na příslušné řídicí procesy (**P1**–**P3**).

V tab. 3.1 lze sledovat výpis jednotlivých verzí datové sady HAI s podrobnostmi, kde jsou jednotlivé datové sady pojmenovány dle měsíce a roku vydání ve formátu "HAI RR.MM".

Tab. 3.1: Jednotlivé verze datové sady HAI s podrobnostmi [58]

Verze datové sady	Normální data			Abnormální data			
	Soubory	**Interval (hodiny)**	**Velikost (MB)**	**Soubory**	**Počet útoků**	**Interval (hodiny)**	**Velikost (MB)**
HAI 22.04	*train1.csv*	26	50,7	*test1.csv*	7	24	48,2
	train2.csv	56	108,9	*test2.csv*	17	23	44,5
	train3.csv	35	66,7	*test3.csv*	10	17,3	33,4
	train4.csv	24	45,7	*test4.csv*	24	36	69,5
	train5.csv	66	125,6	–			
	train6.csv	72	136,8				
	SUM	**279**	**534,4**	**SUM**	**58**	**100,3**	**195,6**
HAI 21.03	*train1.csv*	60	110	*test1.csv*	5	12	22
	train2.csv	63	116	*test2.csv*	20	33	61
	train3.csv	229	245	*test3.csv*	8	30	55
	–			*test4.csv*	5	11	20
				test5.csv	12	26	47
	SUM	**352**	**471**	**SUM**	**50**	**112**	**205**
HAI 20.07	*train1.csv*	86	127	*test1.csv*	28	81	119
	train2.csv	91	98	*test2.csv*	10	42	62
	SUM	**177**	**225**	**SUM**	**38**	**123**	**181**

Scénáře útoku

Všechny scénáře útoků, z hlediska schématu řízení procesu pomocí zpětné vazby, byly navrženy na základě čtyř typů proměnných:

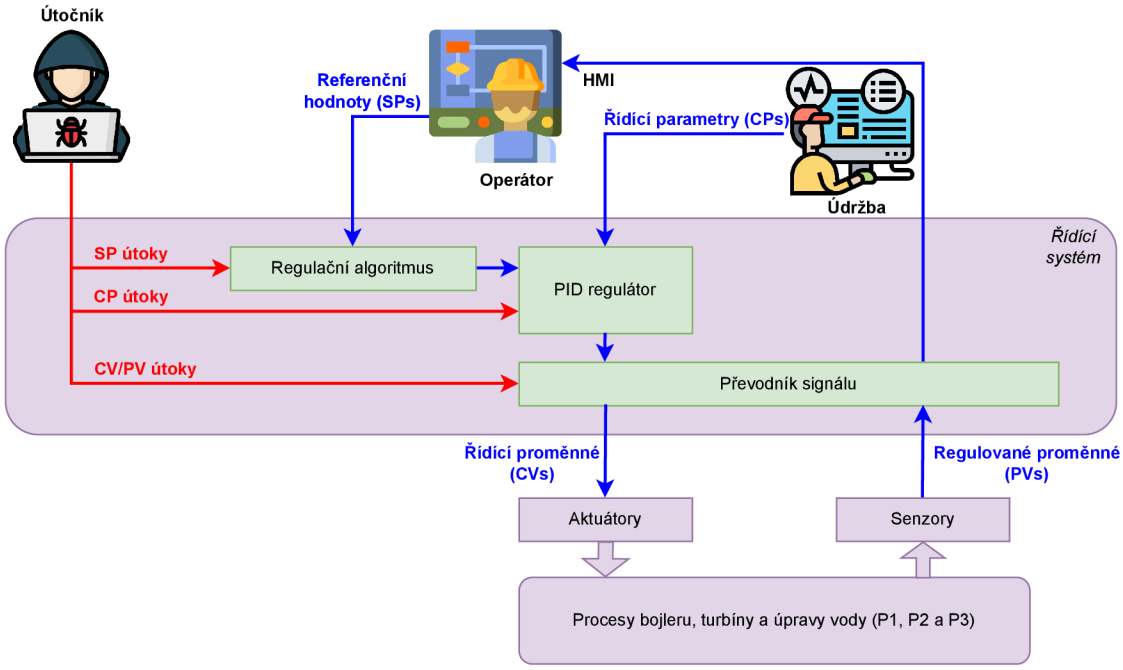
- **Referenční hodnoty** (*Set-points* – SPs) – cílové či požadované hodnoty *regulované proměnné*, kterých se řídicí systém snaží dosáhnout nebo se je snaží udržet.
- **Regulované proměnné (skutečné hodnoty)** (*Process variables* – PVs) – skutečná měření fyzikální veličiny nebo stavu, který je v průmyslovém procesu řízen. PV mohou být např. teplota, tlak, průtok, hladina, pH nebo jakýkoli jiný měřitelný parametr, který je pro řízený proces relevantní.

- **Řídící proměnné** (*Control variables* – CVs) – proměnné, s nimiž se v řídicím systému manipuluje a které se používají k řízení regulační proměnné a k jejímu udržování na požadované hodnotě. Řídicí systém upravuje CV na základě rozdílu mezi SP a PV (tento rozdíl je definován jako tzv. regulační odchylka [2]).
- **Řídící parametry** (*Control parameters* – CPs) – nastavitelné parametry řídicího systému, které určují, jakým způsobem se *řídící proměnné* upravují, aby se *regulovaná proměnná* udržovala na žádané hodnotě. CP se obvykle určují prostřednictvím procesu ladění a mohou zahrnovat proporcionální, integrální a derivační zesílení, jakož i další parametry, které ovlivňují řídicí algoritmus.

Útočník může ovládat všechny proměnné nepřímou manipulací s libovolnými bloky algoritmů ve vestavěných regulátorech, jako je regulační algoritmus, PID regulátor, převodník signálu a další. Útočník tak může v konečném důsledku dosáhnout skrytého útoku na řídicí zařízení a tím může ovlivnit celý proces. Schéma scénáře útoku je vyobrazeno na obr. 3.2.

Při běžném provozu se předpokládá, že obsluha běžně ovládá řídicí zařízení prostřednictvím HMI a že se mění proměnné simulátoru spojené s výrobou energie v simulátoru HIL. Obsluha sleduje regulované proměnné udávané proudovým čidlem zobrazené na HMI a upravuje referenční hodnoty různých řídicích zařízení pro provoz systému. Pomocí plánovače provozních úloh HMI byly pravidelně nastavovány referenční hodnoty a proměnné simulátoru HIL na náhodné nebo předem definované hodnoty v normálním rozsahu, aby se simuloval scénář běžného provozu. Normální rozsahy referenčních hodnot, v nichž byl celý proces stabilní, byly určeny experimentální změnou těchto hodnot. Čtyři regulátory (P1-PC, P1-LC, P1-FC a P1-TC) a dva simulační modely (parní turbínový generátor a přečerpávací vodní generátor) byly automaticky provozovány několikrát denně. Ty byly spouštěny s náhodným zpožděním a bylo dosaženo náhodné hodnoty nebo předem definované hodnoty v rámci normálního provozního rozsahu. Všechny hodnoty SP byly zaznamenány, aby bylo možné zjistit vlastnosti systému [58].

[2] Tato odchylka slouží jako vstup pro regulační algoritmus, který určuje vhodnou akci pro docílení co nejnižší regulační odchylky.



Obr. 3.2: Schéma scénáře útoku [58]

3.2 Porovnání existujících řešení

V rámci datové sady HAI jsou zmíněny projekty pracující na detekci anomálií s pomocí této datové sady. Články [60, 61, 62] k dosažení cíle detekce anomálií užívají strojového učení. Ve [60] je zmíněna soutěž HAICon2020, v rámci které soutěžící pracovali na nejspolehlivějším způsobu detekce anomálií na datové sadě HAI 20.07. HAICon2020 byl první soutěží v Koreji pro modely strojového učení a hlubokého učení, které dokáží odhalit útoky a anomálie tím, že se tyto algoritmy učí pouze z dat z HAI 20.07. Autoři [60] v této soutěži s pomocí jejich řešení získali druhé místo. Navrhli model stacked Bi–LSTM (*Long Short-Term Memory*) ze skupiny rekurentních neuronových sítí (*Recurrent Neural Networks* – RNN). Datové sady pro trénování modelu se skládají z neoznačených normálních dat a vyžadují učení bez učitele. Před samotným učením modelu byla data normalizována tak, aby určité rysy nebyly závislé na jiných. Detekce anomálií je založena na skóre anomálie (anomaly score), které se vypočítá jako rozdíl mezi skutečnými a předpovídanými hodnotami.

4 Praktická část

V této části diplomové práce jsou popsány způsoby, jak lze v praxi použít strojové učení k detekci anomálií, a následně provedení aplikace vhodného řešení (tedy modelu strojového učení) na datovou sadu HAI 22.04.

4.1 Využití strojového učení pro detekci anomálií

Přístupy k detekci anomálií lze rozdělit podle typu dat potřebných k trénování modelu. Ve většině případů použití se očekává, že abnormální vzorky dat představují velmi malé procento celé datové sady. Proto i v případě, že jsou k dispozici označená data, jsou normální vzorky dat snadněji dostupné než případy, kdy data obsahují anomálie [63].

Následuje popis čtyř základních přístupů k učení modelů strojového učení (typy algoritmů strojového učení) [63, 64, 65]:

- **Učení s učitelem** (*Supervised Learning*) – při učení s učitelem se modely strojového učení učí na základě dat a jejich označení. Nejčastěji se užívá tzv. binární klasifikace, tedy pokud existují pouze dva typy označení (např. 0 – normální chování, 1 – anomálie). Cílem algoritmů detekce anomálií s dohledem je začlenit do procesu detekce anomálií znalosti specifické pro danou aplikaci. S dostatečným množstvím normálních a abnormálních příkladů lze úlohu detekce anomálií přeformulovat na klasifikační úlohu, kdy se stroje mohou naučit přesně předpovídat, zda je daný příklad anomálií, či nikoli na základě poskytnutých dat a jejich označení. Nevýhodou tohoto přístupu je nepředvídatelnost nových anomálií. Pokud se model učí z označených historických dat a objeví se nový případ anomálie, může jej vyhodnotit jako normální chování.
  - Algoritmus k-nejbližších sousedů (*k-nearest neighbours*).
  - Rozhodovací stromy (*decision trees*).
  - Lineární regrese (*linear regression*).
  - Metoda podpůrných vektorů (*Support Vector Machines* – SVM).
  - Neuronové sítě (*neural networks*).
- **Učení bez učitele** (*Unsupervised Learning*) – v rámci učení bez učitele modely strojového učení nemají k dispozici příkladové označená data, které by jim umožnily naučit se dopředu, jaká data vykazují probíhající anomálii, a jaká nikoli. Místo toho se učí hledáním souvislostí v poskytnutých datech. Jak již bylo zmíněno, vzorků dat z anomálních situací nebývá mnoho a tím pádem je učení bez učitele z tohoto pohledu vhodnější.
  - Shlukování metodou nejbližších středů (*k-means clustering*).
  - Asociační analýza (*association rule learning*).

- **Semi-supervizované učení** (*Semi-supervised learning*) – přístupy k učení tohoto typu představují kombinaci obou předchozích způsobů (učení s učitelem a učení bez učitele). Jedná se o soubor metod, které využívají velké množství neoznačených dat a malé množství označených dat. Mnoho případů detekce anomálií je pro tento způsob vyhovující, protože je k dispozici velké množství neoznačených normálních příkladů, ze kterých se model může učit a posléze přechází k fázi testování, kde datová sada obsahuje malé množství označených dat.
- **Zpětnovazební učení** (*Reinforcement Learning*) – cílem tohoto algoritmu je využít dedukcí z pozorování (zpětnou vazbu), které jsou získávány z interakce s prostředím. Tímto způsobem se přijímají opatření, která by maximalizovala odměnu nebo minimalizovala riziko. Algoritmus zpětnovazebního učení (nazývaný agent) se průběžně učí z prostředí iterativním způsobem. Agent se přitom učí ze svých zkušeností s prostředím, dokud neprozkoumá celou škálu možných stavů, které mohou nastat. Zpětnovazební učení umožňuje strojům a softwarovým agentům automaticky určovat ideální chování v určité situaci s cílem maximalizovat svůj výkon. K tomu, aby se agent naučil svému chování, je zapotřebí jednoduchá zpětná vazba v podobě odměny (zpětnovazební signál).
  - Q-Learning.
  - Temporal difference.
  - GANs (*Generative Adversarial Networks*).

Pro účely programování v praktické části byl zvolen programovací jazyk Python. Python nabízí stručný a čitelný kód. Algoritmy strojového učení mohou být velice složité, ovšem jednoduchost tohoto programovacího jazyka umožňuje vytvářet spolehlivé modely strojového učení bez nutnosti detailního pochopení toho, jak vše funguje. Efektivní práce s jazykem Python je umožněna také díky existenci nemalého množství knihoven a frameworků (příklady znázorněny níže) právě pro účel strojového učení [66, 68, 69, 70].

- **Analýza a vizualizace dat**:
  - NumPy.
  - SciPy.
  - Pandas.
- **Strojové učení** (*Machine Learning* – ML):
  - Keras.
  - TensorFlow.
  - Scikit-learn.

- **Počítačové vidění** (*Computer Vision* – CVi):
  - OpenCV.
  - SimpleCV.
  - Viso Suite.
- **Zpracování přirozeného jazyka** (*Natural Language Processing* – NLP):
  - NTLK (*Natural Language Toolkit*).
  - CoreNLP.
  - spaCy.

Pro úkol detekce anomálií v průmyslových řídících systémech na datové sadě HAI 22.04 bylo zvoleno učení s učitelem. Všechna data jsou tedy označena (0 – klasický/legitimní provoz, 1 – anomálie). Jak již bylo zmíněno, jedná se o úkol binární klasifikace a pro tento případ jsou vhodné neuronové sítě.

Před samotným popisem modelu je důležité pochopit pár základních pojmů v rámci strojového učení a neuronových sítí [71, 72, 73]:

- **Vzorek** (*sample*) – jeden řádek (záznam) v rámci datové sady.
- **Epocha** (*epoch*) – hyperparametr[1], který určuje, kolikrát model zpracuje trénovací a validační část datové sady.
- **Dávka** (*batch*) – hyperparametr, který definuje počet vzorků, s nimiž se pracuje před aktualizací vnitřních parametrů modelu. Dávku si lze představit jako smyčku for, která iteruje přes jeden nebo více vzorků a provádí předpovědi na základě konkrétního modelu. Na konci každé dávky se předpovědi porovnají s očekávanými výstupními proměnnými a vypočítá se chyba. Z této chyby se pomocí aktualizačního algoritmu (optimalizátoru) model zdokonaluje.
- **Ztrátová funkce** (*loss function/cost function*) – metoda hodnocení algoritmu strojového učení z pohledu, jak dobře si model vede z hlediska předpovědi očekávaného výsledku.
- **Přesnost** (*accuracy*) – Přesnost je jednou z metrik pro hodnocení klasifikačních modelů. Přesnost se vypočítá podílem správných předpovědí a celkovým počtem předpovědí.
- **Váhy** (*weights*) – proměnné parametry konkrétního modelu, které řídí signál (sílu spojení) mezi dvěma neurony. Váhy určují, jak velký vliv bude mít vstup na výstup.
- **Optimalizátor** (*optimizer*) – algoritmus, který upravuje atributy neuronové sítě (zejména váhy). Pomáhá tak snížit celkovou ztrátu a zlepšit přesnost konkrétního modelu.

Vyhodnocení výkonnosti modelu strojového učení má zásadní význam pro určení toho, na kolik je daný model spolehlivý. Matice záměn (*confusion matrix*) je

[1]Parametr, jehož hodnota se používá k řízení procesu učení. Ostatní hodnoty (např. váhy) jsou přímo odvozeny z procesu učení.

užitečným nástrojem pro hodnocení výkonnosti klasifikačních modelů, který poskytuje tabulkové znázornění předpovědí provedených modelem. Je užitečná zejména při práci s nevyváženými soubory dat, kde je rozložení cílové proměnné nerovnoměrné, protože umožňuje určit, jak si model vede v různých situacích. Následuje výpis výkonnostních metrik převzatých z matice záměn [74]

- Nesprávné predikce:
  - **Falešně pozitivní** (*False Positive* – FP) – vztahuje se k počtu případů, kdy model předpověděl pozitivní výsledek, zatímco skutečný výsledek byl negativní (např. vzorek dat byl dle jeho parametrů vyhodnocen jako anomálie, ale jedná se o vzorek z legitimního provozu – falešný poplach).
  - **Falešně negativní** (*False Negative* – FN) – počet případů, kdy model předpověděl negativní výsledek, zatímco skutečný výsledek byl pozitivní (např. vzorek dat byl modelem vyhodnocen jako legitimní, zatímco se jedná o anomálii).
- Korektní predikce:
  - **Pravdivě pozitivní** (*True Positive* – TP) – počet případů, kdy model předpověděl pozitivní výsledek a jednalo se o správnou předpověď (např. model korektně určil anomálii).
  - **Pravdivě negativní** (*True Negative* – TN) – počet případů, kdy model předpověděl negativní výsledek a jednalo se o správnou předpověď (např. model korektně určil legitimní provoz).

Tyto metriky mají zásadní význam pro hodnocení výkonnosti klasifikačního modelu, protože umožňují pochopit, kde model dělá chyby. Například model s vysokým počtem falešně pozitivních výsledků může nesprávně identifikovat příliš mnoho případů jako pozitivní, což může mít v závislosti na aplikaci závažné důsledky. Na druhou stranu model s vysokým počtem falešně negativních výsledků může přehlížet příliš mnoho případů, které jsou ve skutečnosti pozitivní, což může mít rovněž vážné důsledky. Tyto dva případy (nadměrné FP, či nadměrné FN) jsou nejčastějšími problémy modelů strojového učení pro případy klasifikace.

Pro detailnější analýzu výkonnosti modelů strojového učení existují další metriky, které detailněji popisují, jak je model účinný. Následující metriky se běžně používají v klasifikačních úlohách k hodnocení výkonnosti modelů strojového učení. Poskytují přehled o schopnosti modelu správně klasifikovat různé třídy a pomáhají identifikovat oblasti pro zlepšení daného modelu [75, 76]:

- **Přesnost** (*Accuracy*) – podíl korektních předpovědí pro testovací datovou sadu. Přesnost lze spočítat vydělením počtu korektních předpovědí počtem všech předpovědí;

$$přesnost = \frac{správné\ předpovědi}{všechny\ předpovědi} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Preciznost** (*Precision*) – podíl korektně předpovězených anomálií vzhledem k nesprávným předpovědím, tedy kolik označených anomálií je relevantních;

$$preciznost = \frac{TP}{TP + FP}$$

- **Senzitivita** (*Recall*) – podíl korektně předpovězených anomálií vzhledem anomáliím nesprávně vyhodnoceným jako legitimní provoz, tedy kolik relevantních anomálií je označených;

$$senzitivita = \frac{TP}{TP + FN}$$

- **Specificita** (*Specificity*) – podíl dat správně označených jako legitimní vzhledem k legitimnímu provozu nesprávně označenému jako anomálie, tedy jaké množství legitimního provozu je správně označeno;

$$specificita = \frac{TN}{TN + FP}$$

- **F-míra** (*F-score*) – tato metrika je užitečná k výpočtu úspěšnosti konkrétního modelu, pokud se hodnoty preciznosti a senzitivity výrazně liší, jedná se o jejich harmonický průměr.

$$F\text{-}míra = 2 * \frac{preciznost * senzitivita}{preciznost + senzitivita}$$

V závislosti na těchto metrikách se následně lze zaměřit na úpravu některých parametrů modelu – hyperparametrů. Tato činnost se nazývá tzv. *Hyperparameter Optimization*, či *Fine-tuning*. Obecně se modely strojového učení skládají ze dvou typů parametrů [77]:

- **Obecné parametry modelu** (*Model parameters*) – jedná se o parametry, které se učí během procesu trénování modelu. Těmito parametry jsou váhy mezi jednotlivými vrstvami neuronové sítě. V průběhu trénování se tyto váhy neustále upravují, aby model lépe popisoval vstupní data a dokázal správně klasifikovat nová data. U jiných typů modelů se mohou objevit jiné obecné parametry, které jsou také specifické pro daný typ modelu (např. váhy (*weights*)).
- **Hyperparametry** (*Hyperparemeters*) – jedná se o parametry, které nemají být učeny během trénování, ale jsou nastavovány uživatelem před samotným trénováním modelu. Tyto parametry ovlivňují proces učení a výkonnost modelu. Mezi tyto parametry mohou patřit například počet vrstev v neuronové síti, velikost jednotlivých vrstev, míra učení, počet epoch, velikost dávek (*batch size*) a další. Správné nastavení hyperparametrů může zásadně ovlivnit výkon a schopnost modelu se učit. Proto je důležité tyto parametry pečlivě volit a vybírat takové hodnoty, které povedou k nejlepším výsledkům.

Pomocí ladění hyperparametrů lze následně model zkoušet s různými variacemi parametrů pro docílení nejlepších možných výsledků.

Ke zlepšení výsledků různých modelů strojového učení lze aplikovat různé úpravy datové sady před samotným učením.Tyto úpravy datové sady jsou důležité z několika důvodů. Proměnné, které jsou měřeny v různých měřítkách, mohou mít různý vliv na výsledky a mohou způsobovat zkreslení, což může zhoršit konečné výsledky modelů strojového učení. Standardizace a normalizace umožňují zajistit, že všechny hodnoty jsou na stejné stupnici. Přeškálování zase umožňuje změnu jednotek, aby byly data srovnatelná a snadno interpretovatelná. Tyto úpravy jsou důležité pro dosažení nejlepších výsledků z modelů strojového učení [78].

- **Standardizace** (*Standardizing*) – odečtení míry umístění a dělení mírou měřítka. Pokud například vektor obsahuje náhodné hodnoty s Gaussovým rozdělením, lze odečíst průměr a vydělit směrodatnou odchylkou, čímž je získána "standardní normální" náhodná veličina se střední hodnotou 0 a směrodatnou odchylkou 1. Proměnné, které jsou měřeny v různých měřítkách, nepřispívají k analýze se stejnou důležitostí a tento fakt má tendenci způsobovat zkreslení. Pokud tedy v rámci datové sady existuje řada různých parametrů, kde jeden může mít rozsah 0 až 1 a druhý 0 až 100 000. V takovém případě může mít algoritmus strojového učení problém s přidělováním vah jednotlivých prvků.
- **Normalizace** (*Normalizing*) – nejčastěji znamená dělení normou vektoru. Často také znamená změnu měřítka podle minima a rozsahu vektoru, aby všechny prvky ležely mezi 0 a 1, čímž se všechny hodnoty číselných sloupců v souboru dat dostanou do společného měřítka. Podobně, jako u standardizace, je cílem normalizace změnit hodnoty číselných sloupců v datové sadě na společnou stupnici, aniž by došlo ke zkreslení rozdílů v rozmezích hodnot. Pro strojové učení nevyžaduje každá datová sada normalizaci, je nutná pouze tehdy, když mají prvky různé rozsahy.
- **Přeškálování** (*Rescaling*) – přičtení nebo odečtení konstanty a následné vynásobení nebo dělení konstantou, což se provádí při změně měrných jednotek dat.

K úpravě datové sady je vhodná knihovna Pandas a ke konečnému vykreslení grafů knihovna Matplotlib. Pandas je užitečná knihovna pro úpravu a manipulaci s datovými sadami, zatímco Matplotlib je knihovna, kterou lze použít pro vytváření vizualizací a grafů. Obě knihovny jsou užitečné k analýze a prezentaci výsledků [67].

4.2 Statistická analýza

K samotnému řešení problému binární klasifikace (tedy určení, zda se jedná o normální situaci, či anomálii) datové sady HAI 22.04 bylo přistoupeno k řešení pomocí tří specifických přístupů:

- **Strojové učení s učitelem** – pro tento přístup existuje široká škála algoritmů, které jsou vhodné pro řešení problému binární klasifikace. Jelikož datová sada HAI 22.04 nezahrnuje v trénovací části anomální data, je v tomto případě třeba pracovat pouze s csv soubory určenými pro testování, konkrétně *test1.csv*, *test2.csv*, *test3.csv* a *test4.csv*.
- **Neuronová síť** – neuronovou síť lze vytvořit pomocí knihovny Keras. Jak je zmíněno v [68], Keras umožňuje snadno navrhnout neuronovou síť a provádět s ní různé operace. K řešení problému binární klasifikace lze také přistoupit s pomocí knihovny Tensorflow [79]. TensorFlow poskytuje jednoduché rozhraní, které umožňuje vytvářet a trénovat modely určené k binární klasifikaci. Zároveň lze tyto modely přizpůsobit pro optimální výkon a škálovatelnost.
- **Strojové učení bez učitele** – v tomto řešení je možné použít všechny trénovací datové sady bez označení, zda se jedná o legitimní data, či anomálii. Cílem v této části je naučit model strojového učení, aby byl schopen detekovat anomální chování na základě rozdílnosti dat, a následně jej otestovat na testovacích datových sadách a sledovat, zda dokáže efektivně detekovat anomálie.

4.2.1 Strojové učení s učitelem

V této části diplomové práce byly vybrány soubory datové sady HAI 22.04, které zahrnují jak legitimní data, tak i data zachycená po dobu probíhajících útoků. Následuje výpis počtů jednotlivých vzorků dat vzhledem ke konkrétním datovým sadám:

- **test1.csv**:
  - Legitimní data – 85515 (98,97 %)
  - Anomální data – 885 (1,03 %)
- **test2.csv**:
  - Legitimní data – 79919 (96,39 %)
  - Anomální data – 2881 (3,61 %)
- **test3.csv**:
  - Legitimní data – 58559 (93,84 %)
  - Anomální data – 3841 (6,16 %)
- **test4.csv**:
  - Legitimní data – 125177 (96,59 %)
  - Anomální data – 4423 (3,41 %)

Pro tento přístup byla vybrána jako trénovací datová sada *test4.csv* z toho důvodu, že tento soubor obsahuje nejvíce vzorků dat, přičemž část této datové sady (konkrétně 20 %) slouží pro validaci. Pro testování byly zvoleny datové sady *test1.csv*, *test2.csv* a *test3.csv*.

Pro učení s učitelem bylo implementováno celkově 11 různých modelů strojového učení, kde většina pochází z knihovny `Scikit-learn` [80], a zbytek z unikátních knihoven: `XGBoost`, `LightGBM` a `CatBoost` [81, 82, 83]:

- **Logistic Regression**
- **Support Vector Machines (SVM)**
- **Decision Trees**
- **Random Forest**
- **Naive Bayes**
- **K-Nearest Neighbor**
- **Gradient Boosting**
- **AdaBoost**
- **XGBoost**
- **LightGBM**
- **CatBoost**

Všechny operace s datovou sadou a následné trénování, validace a testování jednotlivých modelů strojového učení byly provedeny ve vývojovém prostředí `Google Colab`.

Předzpracování datové sady

Po importování potřebných knihoven proběhlo nahrání všech datových sad, přičemž každá z nich obsahuje 88 sloupců (86 datových bodů zmíněných v tabulkách A.1 a A.2, časová značka a hodnota označující, zda probíhá útok – "0"pro normální provoz, nebo "1", značící útok). Bez jakékoliv analýzy důležitosti hodnot z jednotlivých datových bodů bylo možné vyloučit jakákoliv data, která jsou pro všechny vzorky dat neměnná, tedy pokud je hodnota pro daný datový bod konstantní (před tímto odstraněním konstantních dat bylo nutné všechny datové sady spojit do jedné pro případ, že by v rámci jednotlivých souborů existovaly rozdíly v rámci konstantních dat). Taktéž byl odstraněn sloupec s časovou značkou, jelikož byla vybrána data, která na sebe z hlediska času navazují. Po této operaci byly datové sady zredukovány na 69 sloupců a následně rozděleny zpět do svých původních velikostí (dle počtu řádků). Následně byly určeny parametry pro rozdělení datových sad na trénovací, validační a testovací, kde soubor *test4.csv* byl rozdělen na trénovací a validační datovou sadu v poměru 80:20, a zbytek, tedy *test1.csv*, *test2.csv* a *test3.csv*, byl určen jako testovací datové sady. Tímto byly datové sady dle stanoveného poměru

rozděleny na části s daty, ze kterých se modely učí (**trénování**) a na ty, kde modely ověřují správnost učení (**validace**). Zbylé části slouží ke zjištění správnosti předpovědí (**testování**). Před samotným učením modelů bylo taktéž nutné provést rozdělení trénovacích, validačních a testovacích dat na predikční a cílové proměnné. Predikční proměnné (pouze data z jednotlivých bodů bez označení, zda se jedná o útok, či ne) slouží pouze k predikcím, kdežto cílové proměnné jsou určeny k ověření správnosti těchto predikcí. Všechny uvedené operace byly naprogramovány v sešitu `Google Colab` a jsou znázorněny v obr. 4.1.

```
# Načtení datových sad
train = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/datasets/test4.csv')
test1 = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/datasets/test1.csv')
test2 = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/datasets/test2.csv')
test3 = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/datasets/test3.csv')

# Sjednocení datových sad pro lepší manipulaci
train_test = pd.concat([train, test1, test2, test3], axis=0)

# Odstranění sloupce s časovou značkou
train_test =  train_test.drop('timestamp', axis=1)

# Odstranění sloupců s konstantními hodnotami
train_test = train_test.loc[:, (train_test != train_test.iloc[0]).any()]

# Rozdělení datových sad po počátečních úpravách
train = train_test.iloc[:train.shape[0], :]
test1 = train_test.iloc[train.shape[0]:train.shape[0]+test1.shape[0], :]
test2 = train_test.iloc[train.shape[0]+test1.shape[0]:train.shape[0]+test1.shape[0]+test2.shape[0], :]
test3 = train_test.iloc[train.shape[0]+test1.shape[0]+test2.shape[0]:, :]

# Oddělení predikčních proměnných od cílové proměnné u trénovací datové sady
x = train.drop('Attack', axis=1)
y = train['Attack']

# Rozdělení trénovací datové sady na trénovací a validační v poměru 80:20
x_train, x_valid, y_train, y_valid = train_test_split(x, y, test_size=0.2, shuffle=False)

# Vytvoření predikčních proměnných pro testovací datovou sadu
x_test1 = test1.drop('Attack', axis=1)
x_test2 = test2.drop('Attack', axis=1)
x_test3 = test3.drop('Attack', axis=1)

# Vytvoření cílových proměnných pro testovací datovou sadu
y_test1 = test1['Attack']
y_test2 = test2['Attack']
y_test3 = test3['Attack']
```

Obr. 4.1: Předzpracování datové sady a příprava k procesu učení a testování

Trénování, validace a testování jednotlivých modelů

Před samotným procesem trénování, validace a testování bylo nutné importovat všechny modely z patřičných knihoven. Vytvořený kód inicializuje prázdný slovník nazvaný *models* a přidává do něj instance různých klasifikačních modelů z knihoven scikit-learn, XGBoost, LightGBM a CatBoost (obr. 4.2).

```
models = {}

# Logistic Regression
from sklearn.linear_model import LogisticRegression
models['Logistic Regression'] = LogisticRegression()

# Support Vector Machines
from sklearn.svm import LinearSVC
models['Support Vector Machines'] = LinearSVC()

# Decision Trees
from sklearn.tree import DecisionTreeClassifier
models['Decision Trees'] = DecisionTreeClassifier()

# Random Forest
from sklearn.ensemble import RandomForestClassifier
models['Random Forest'] = RandomForestClassifier()

# Naive Bayes
from sklearn.naive_bayes import GaussianNB
models['Naive Bayes'] = GaussianNB()

# K-Nearest Neighbors
from sklearn.neighbors import KNeighborsClassifier
models['K-Nearest Neighbor'] = KNeighborsClassifier()

# Gradient Boosting Classifier
from sklearn.ensemble import GradientBoostingClassifier
models['Gradient Boosting'] = GradientBoostingClassifier()

# AdaBoost Classifier
from sklearn.ensemble import AdaBoostClassifier
models['AdaBoost'] = AdaBoostClassifier()

# XGBoost Classifier
from xgboost import XGBClassifier
models['XGBoost'] = XGBClassifier()

# # LightGBM Classifier
from lightgbm import LGBMClassifier
models['LightGBM'] = LGBMClassifier()

# CatBoost Classifier
from catboost import CatBoostClassifier
models['CatBoost'] = CatBoostClassifier(verbose=False)
```

Obr. 4.2: Inicializace jednotlivých modelů strojového učení

Každý model byl pro počáteční testování vytvořen s výchozími hyperparametry a uložen do slovníku *models* s identifikátorem modelu (názvem) jako klíčem. Účelem uložení modelů do slovníku je umožnit snadnou iteraci nad modely. Později v kódu lze iterovat nad modely po jednom a každý model lze učit a vyhodnocovat na stejné datové sadě. To umožňuje efektivní porovnání výkonnosti jednotlivých modelů.

Před započetím samotného procesu učení a testování jednotlivých modelů byly definovány potřebné proměnné pro ukládání výkonnostních metrik modelu, a to následující:

- **Přesnost** (*Accuracy*)
- **Preciznost** (*Precision*)
- **Senzitivita** (*Recall*)
- **F–míra** (*F–score*)
- **Matice záměn** (*Confusion Matrix*)
- **TN, FP, FN, TP**

Tyto výkonnostní metriky se postupně ukládají do patřičných slovníků při trénování každého modelu. Jelikož bylo k testování modelů přistoupeno taktikou, kde jsou predikce určeny pro každou ze tří testovacích datových sad zvlášť, bylo z tohoto důvodu nutné vytvořit od každé proměnné vytvořit tři typy. Kód pro tuto část procesu je vyobrazen na obr. 4.3.

Pro získání širšího rozsahu výsledků pro hodnocení výkonnosti jednotlivých modelů byl použit stejný postup, ovšem s normalizovanými daty pomocí knihoven `StandardScaler` a `MinMaxScaler`. Tímto byla data škálována na určitý rozsah, aby bylo možné snadněji porovnávat různé funkce. Knihovna `StandardScaler` škáluje data tak, aby měla nulový průměr a jednotkový rozptyl, zatímco knihovna `MinMaxScaler` škáluje data na rozsah mezi hodnotami 0 a 1. Normalizace dat může pomoci zlepšit přesnost modelů strojového učení tím, že sníží vliv dat, které mají různé škály a rozsahy. Byly tedy provedeny tři různé scénáře:

- **1. scénář** – data bez úpravy.
- **2. scénář** – data upravená pomocí `StandardScaler` (standardizace).
- **3. scénář** – data upravená pomocí `MinMaxScaler` (normalizace).

```
# Vytvoření proměnných pro všechny výkonnostní metriky
accuracy_1, precision_1, recall_1, f1_1, cm_1 = {}, {}, {}, {}, {}
accuracy_2, precision_2, recall_2, f1_2, cm_2 = {}, {}, {}, {}, {}
accuracy_3, precision_3, recall_3, f1_3, cm_3 = {}, {}, {}, {}, {}

# Cyklus pro postupné iterování nad všemi užitými modely
for key in models.keys():

    # Provedení cross-validace na validační datové sadě
    scores = cross_val_score(models[key], x_valid, y_valid, cv=5, scoring='accuracy')

    # Trénování modelů
    models[key].fit(x_train, y_train)

    # Testování modelů (provádění predikcí)
    predictions_1 = models[key].predict(x_test1)
    predictions_2 = models[key].predict(x_test2)
    predictions_3 = models[key].predict(x_test3)

    # Výpočet výkonnostních metrik pro jednotlivé testovací datové sady
    accuracy_1[key] = accuracy_score(predictions_1, y_test1)
    precision_1[key] = precision_score(predictions_1, y_test1)
    recall_1[key] = recall_score(predictions_1, y_test1)
    f1_1[key] = f1_score(predictions_1, y_test1)
    cm_1[key] = confusion_matrix(predictions_1, y_test1)

    accuracy_2[key] = accuracy_score(predictions_2, y_test2)
    precision_2[key] = precision_score(predictions_2, y_test2)
    recall_2[key] = recall_score(predictions_2, y_test2)
    f1_2[key] = f1_score(predictions_2, y_test2)
    cm_2[key] = confusion_matrix(predictions_2, y_test2)

    accuracy_3[key] = accuracy_score(predictions_3, y_test3)
    precision_3[key] = precision_score(predictions_3, y_test3)
    recall_3[key] = recall_score(predictions_3, y_test3)
    f1_3[key] = f1_score(predictions_3, y_test3)
    cm_3[key] = confusion_matrix(predictions_3, y_test3)

    # Extrahování hodnot TN, FP, FN, TP z matice záměn
    tn_1, fp_1, fn_1, tp_1 = cm_1[key].ravel()
    tn_2, fp_2, fn_2, tp_2 = cm_2[key].ravel()
    tn_3, fp_3, fn_3, tp_3 = cm_3[key].ravel()
```

Obr. 4.3: Trénování, validace a testování jednotlivých modelů

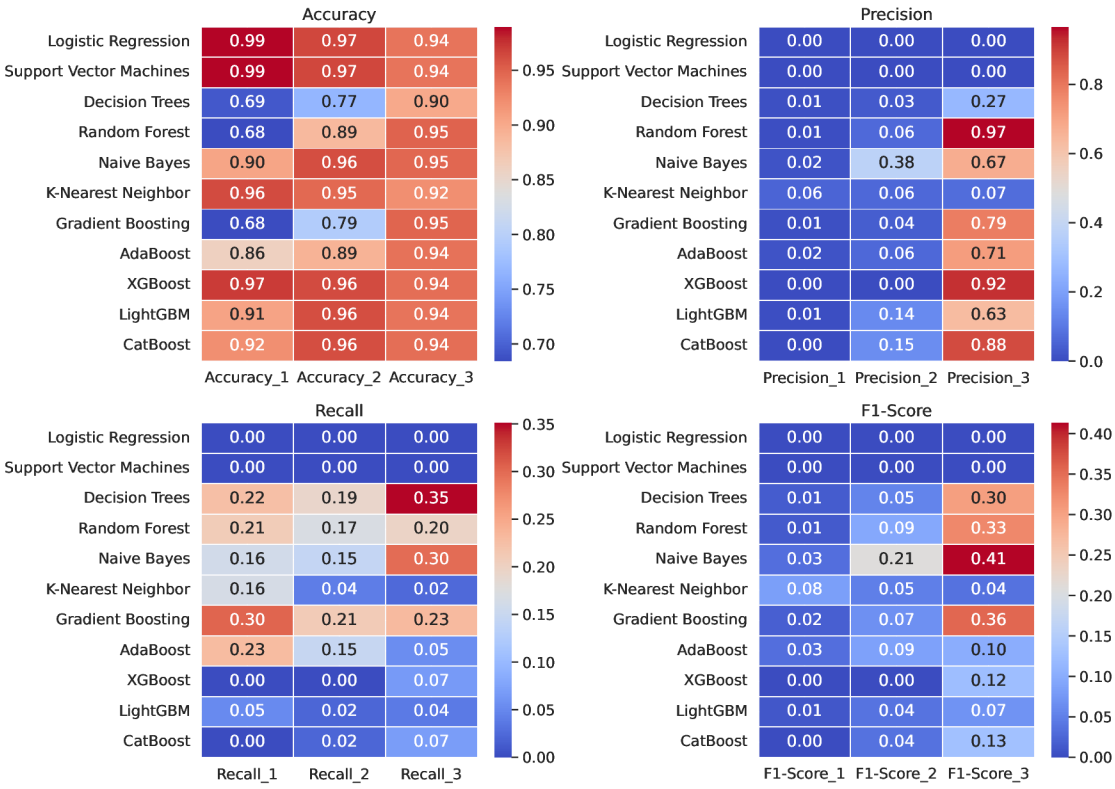
Výsledky modelů strojového učení s učitelem

V první řadě byl proveden rozbor výsledků řešení, kde nebyly datové sady normalizovány (1. scénář). Dle prvotních výsledků lze sledovat, že nějaké modely mají díky nízkému počtu vzorků dat označených "1"(anomálie) tendenci označit všechna data "0", tedy predikovat je jako legitimní. Tímto je sice dosaženo vysoké přesnosti modelu, ale zároveň se jedná o hrubou chybu v rámci binární klasifikace, jelikož jsou modely prakticky nefunkční. V tomto případě se jedná o modely "Logistic Regression"a "Support Vector Machines". Důvodů pro tyto výsledky může být několik:

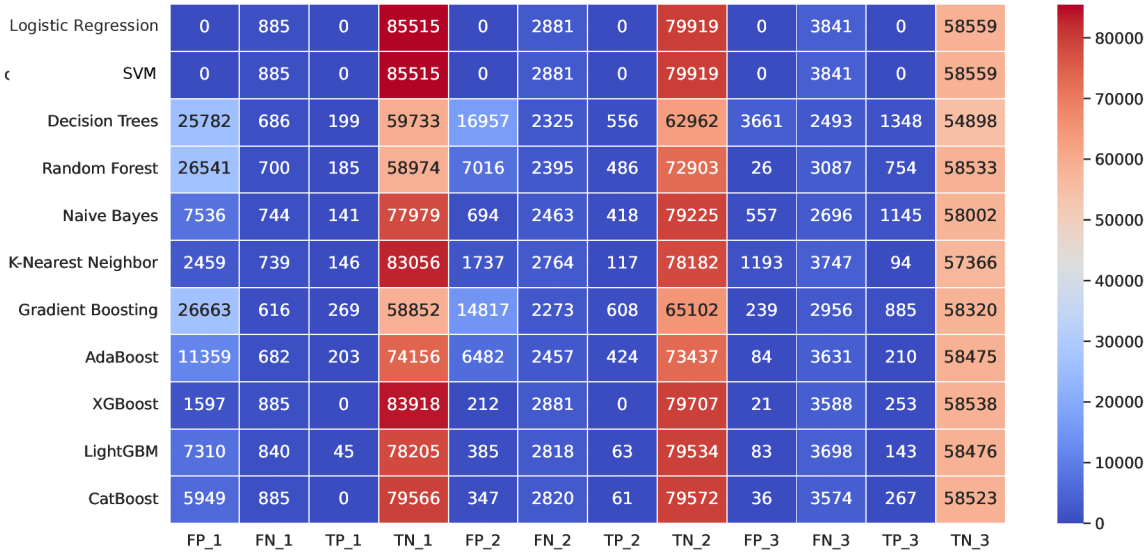
- **Problémy s daty** – vstupní data nemusí být pro tyto algoritmy vhodná. V tomto případě se jedná o nevyváženost datových sad a tím pádem je pro tyto modely jednodušší predikovat všechna data jako legitimní, protože tím dosáhnou nejvyšší přesnosti.
- **Ladění hyperparametrů** – hyperparametry algoritmů nejsou pro daná data správně vyladěny.
- **Overfitting** – pokud se modely nadměrně přizpůsobují trénovacím datům, může při předložení nových, neznámých dat předpovídat všechny cílové proměnné jako "0".

V rámci analýzy výsledků byly vygenerovány dva grafy pomocí knihoven Matplotlib a Seaborn. Pro lepší přehlednost výsledků bylo využito grafu typu "heatmap", který mění barvy jednotlivých kolonek na základě jejich hodnot. První graf byl sestaven pomocí výkonnostních metrik přesnost, preciznost, senzitivita a f1-skóre (viz obr. 4.4) a druhý graf na základě naměřených hodnot matice záměn (obr. 4.5). Je důležité podotknout, že v rámci těchto grafů číselné označení metrik (např. "Accuracy_1", "Accuracy_2"a "Accuracy_3") neindikuje různé scénáře (dle normalizace dat), ale značí datové sady, ze kterých po predikci modelů byly tyto metriky získány, tedy *test1.csv*, *test2.csv*, respektive *test3.csv*.
Při analýze dotyčných grafů lze konstatovat několik faktů:

- **Vysoká přesnost** – až na výjimky dosahuje přesnost všech algoritmů velmi vysokých hodnot. Tento fakt ovšem neznačí opravdovou úspěšnost modelů, jak již bylo řečeno dříve. Je třeba také brát v potaz ostatní výkonnostní metriky.
- **Korelace mezi prvky matice záměn** – některé modely vykazují přímou závislost TP na FN (tedy čím více TP, tím více FP). Ve výsledku to znamená, že model sice dokáže správně predikovat více anomálií, ale tím pádem vychází i více falešně pozitivních výsledků.
- **Neoptimální výsledky** – žádný z testovaných algoritmů strojového učení nedosahoval optimálních výsledků. I přes poměrně vysoké procento FN u všech modelů lze konstatovat, že nejúspěšnější byly algoritmy Decision Trees, Random Forest, Naive Bayes a Gradient Boosting.



Obr. 4.4: Výsledky modelů (výchozí řešení) – výkonnostní metriky



Obr. 4.5: Výsledky modelů (výchozí řešení) – matice záměn

V rámci 2. scénáře (standardizace datových sad) a 3. scénáře (normalizace datových sad) bylo ve všech případech dosaženo horších výsledků, než při práci s neupravenými daty. Důvodem pro tyto výsledky může být nevhodnost datové sady na takové úpravy. Datová sada HAI obsahuje velké množství datových bodů, a zároveň existuje vysoká rozdílnost rozsahů jednotlivých datových bodů. Z tohoto důvodu může být v tomto případě aplikování standardizace, či normalizace nevhodné. Ve výsledcích modelů trénovaných a testovaných na standardizovaných, a normalizovaných datech dochází ke značně vyšší četnosti předpovědí typu FP, než u modelů, které pracovaly s neupravenými daty.

Optimalizace výsledků

Modifikacemi vytvořeného zdrojového kódu lze dosažené výsledky optimalizovat. Bylo provedeno několik různých přístupů [2] pro zlepšení konečných výsledků detekce anomálií. Následuje výpis všech uskutečněných scénářů, které byly provedeny pro optimalizaci výsledků, komentář ke každému z nich:

- **Vyvážení trénovací datové sady** – v tomto přístupu bylo realizováno vyvážení trénovací datové sady pomocí knihovny `imblearn` [84] (konkrétně funkce RandomOverSampler). RandomOverSampler určí vzorky minoritních tříd (v tomto případě anomálie –> označení "1") v datové sadě, které mají méně případů v porovnání s majoritní třídou (v tomto případě legitimní data –> označení "0"). Po určení těchto tříd náhodně vybere vzorky z menšinové třídy a duplikuje je, aby v datové sadě zvýšil jejich četnost. Tento proces se opakuje, dokud se počet vzorků v minoritní třídě neshoduje s počtem vzorků v majoritní třídě. Pokud se tedy provede vyvážení trénovací datové sady (*test4.csv*), výsledkem bude 125 177 vzorků dat v obou třídách "0", i "1". Výsledkem vyvážení bylo zhoršení přesnosti většiny modelů. Jediný případ, kde došlo k minimálnímu zlepšení, byl algoritmus Naive Bayes. V ostatních případech docházelo k rapidnímu nárůstu předpovězených FP, a v některých i k poklesu TP. Důvodem pro tyto výsledky může být fakt, že vzorky označené "1"jsou v datové sadě zastoupeny v tak malé míře, že vyvážením trénovací datové sady je dosaženo spíše zhoršení, než zlepšení predikcí. Byl vyzkoušen i scénář, kde bylo v trénovací datové sadě provedeno náhodné vyvážení nahrazením vzorků dat označených "0"vzorky s označením "1"a tedy vznikla trénovací datová sada s 64 800 datovými prvky z klasického provozu a 64 800 anomálními prvky. Tento scénář ovšem výsledky nijak nevylepšil, pouze adekvátně snížil čas učení jednotlivých modelů.

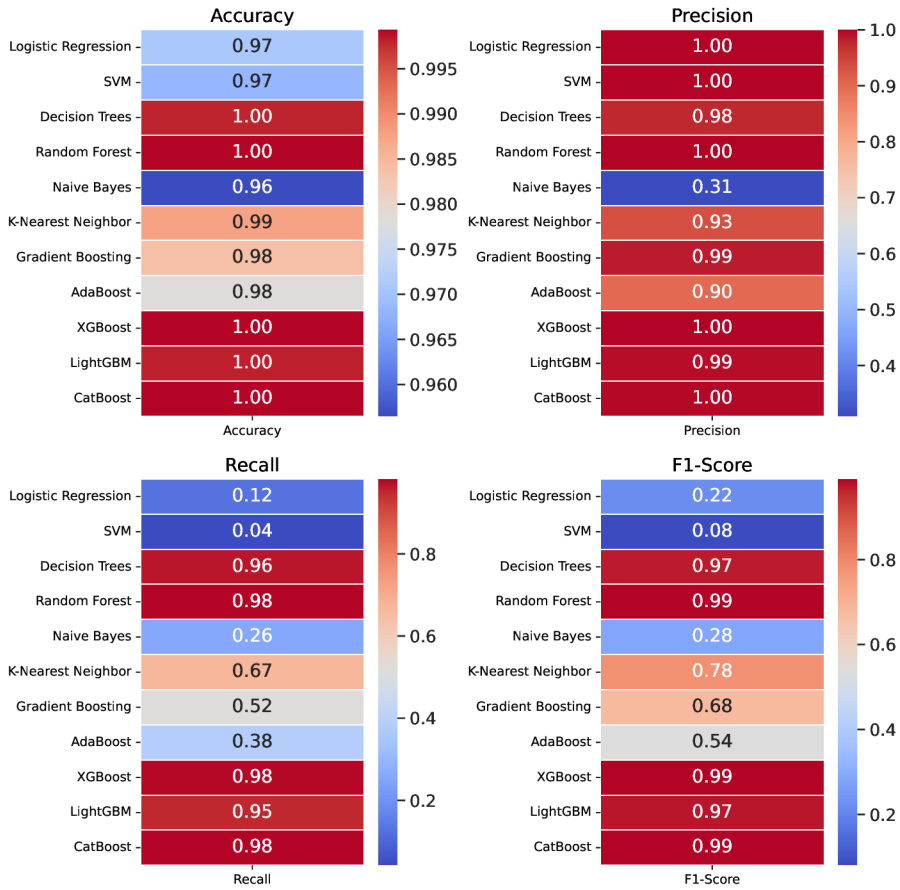
[2] Nutno podotknout, že všechny následující scénáře jsou vytvořeny s pomocí výchozího zdrojového kódu se specifickými úpravami dle konkrétního scénáře.

- **Změna poměrů datových sad** – změnou poměrů datových sad je v tomto případě myšleno užití více datových sad pro učení modelů (konkrétně *test1.csv*, *test2.csv* a *test4.csv*) a jejich otestování na pouze jedné datové sadě (*test3.csv*). Tento přístup byl realizován bez zamíchání a vyvážení datových sad. Při srovnání výsledků s přístupy v obrázcích 4.4 a 4.5 lze konstatovat, že zvýšení obsahu trénovací datové sady (a tedy i validační) oproti testovací datové sadě mělo za následek velice srovnatelné výsledky s výchozím řešením (konkrétně metriky označené "_3", jelikož šlo o testování na stejné datové sadě *test3.csv*). Výsledky byly v tomto případě mnohdy i horší, tudíž lze spekulovat o přeučení modelů přílišným množstvím vzorků dat.
- **Ladění hyperparametrů** – ladění hyperparametrů proces výběru nejlepší sady hyperparametrů pro konkrétní model strojového učení. Jak již bylo zmíněno, tyto parametry se neučí z dat, ale jsou nastaveny uživatelem před trénováním modelu (či přednastaveny ve výchozím nastavení modelu). Určují různé aspekty algoritmu učení a tím pádem ovlivňují výkon a chování modelu. Pomocí ladění hypermarametrů lze dosáhnout lepších, ale i horších výsledků (výběr nevhodných nebo neoptimálních hodnot hyperparametrů může vést ke špatnému výkonu modelu). Tento přístup je problematický z hlediska nalezení optimálních výsledků, protože je časově náročný z hlediska vhodnosti konkrétních parametrů na datové sadě HAI pro dosažení optimálních výsledků. Z tohoto důvodu byly redukovány užité modely na Decision Trees, Random Forest, Naive Bayes, K–Nearest Neighbours, Gradient Boosting a XGBoost. Pomocí ladění hyperparametrů ovšem nebylo dosaženo optimálních výsledků, opět nastala situace, kdy výsledky byly srovnatelné, či horší v porovnání s výchozím řešením.
- **Odstranění redundantních datových bodů (sloupců)** – cílem tohoto přístupu bylo vyfiltrovat všechny nedůležité datové body z hlediska posouzení do jaké míry (a jestli vůbec) jsou hodnoty pro jednotlivé vzorky odlišné pokud se jedná o legitimní provoz, nebo anomálii. Jak již bylo zmíněno, z datové sady byly odstraněny datové body s konstantními hodnotami. Následně byla provedena analýza hodnot zbylých 68 datových bodů s cílem zjistit, jaké z těchto datových bodů jsou pro účel strojového učení zbytečné. Na obrázcích B.1, B.2, B.3 a B.4 lze sledovat vykreslení všech hodnot jednotlivých datových bodů datových sad *test1.csv*, *test2.csv*, *test3.csv* a *test4.csv*, přičemž v každém grafu jdou vidět hodnoty ze všech 4 datových sad (černé, vertikální dělící linky slouží k oddělení jednotlivých datových sad v grafech). Navíc jsou pro lepší orientaci odděleny vzorky legitimních (**modrá barva**), a anomálních (**červená barva**) dat. Při pohledu na obr. B.1 lze například konstatovat, že v datovém bodu "P1_B2004" jsou všechny vzorky dat ve velké míře s konstantní

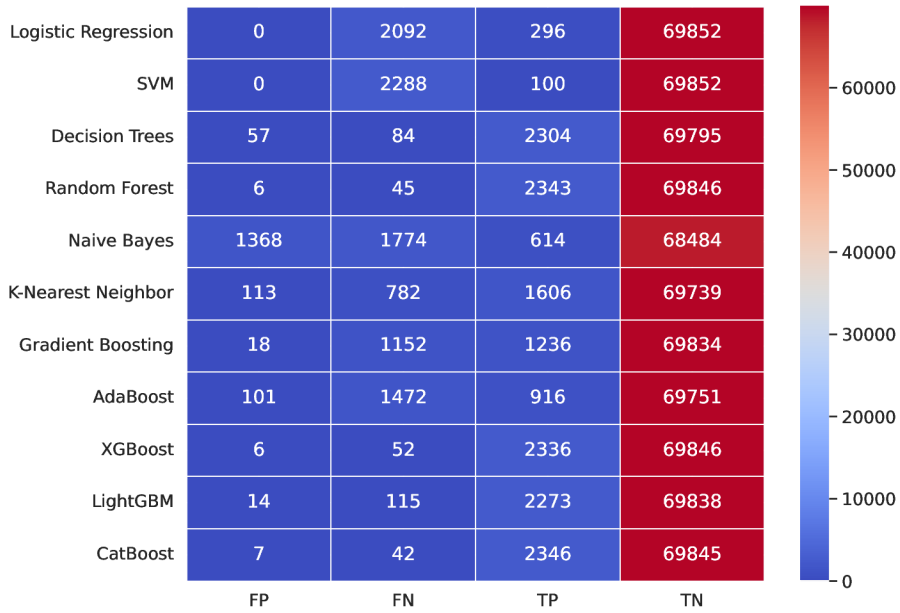
hodnotou, a navíc se hodnoty anomálních dat nijak neliší od těch legitimních. Oproti tomu, v datovém bodu "P1_FT01" lze s jistotou říci, že hodnoty nejsou konstantní, přičemž lze jasně určit, že hodnoty anomálních dat se vymykají normě, ovšem ne ve všech případech. Nakonec jsou přítomny takové datové body, u kterých se pohybují hodnoty legitimních a anomálních vzorků dat ve stejných rozsazích (např. "P1_B2016"). Datových bodů tohoto typu je vícero a jsou problematické, protože z nich nelze na první pohled určit, zda mohou mít vliv na učení konkrétního algoritmu. Anomální vzorky dat jsou sice ve stejném rozsahu, jako ty legitimní, ale může u nich například docházet k rychlejším změnám hodnot. Na základě analýzy vykreslených grafů byly vyhotoveny 3 přístupy k odstraňování redundantních datových bodů:

  - 1. přístup – ponechání datových bodů typu "P1_FT01" a "P1_B2016", tedy vyfiltrování všech datových bodů, u kterých jsou hodnoty z velké části konstantní, a anomální hodnoty se neliší od těch legitimních. Výsledkem je 50 datových bodů.
  - 2. přístup – odstranění všech datových bodů, kromě takových, u kterých se anomální hodnoty zřetelně liší od těch legitimních (např. "P1_FT01"). Výsledkem je 17 datových bodů.
  - 3. přístup – v tomto přístupu byly ponechány pouze ty datové body, na které byly provedeny cílené scénáře útoků (tabulky A.3 a A.4). Výsledkem je 20 datových bodů.

  Z vyjmenovaných situací měl nejlepší výsledky 2. přístup, který dosahoval srovnatelných, u nějakých modelů i lepších výsledků, než výchozí řešení.
- **Promíchání dat** – náhodným promícháním dat může z hlediska strojového učení pomoci, a proto bylo přistoupeno k rozdělení datové sady pomocí funkce "train_test_split", a to v poměru 80 % trénovací data a 20 % testovací data. Tento způsob se jevil nejefektivnějším z hlediska účinnosti modelů v předpovědích anomálií s velice vysokou úspěšností. Výsledky pro data neupravená pomocí normalizace a standardizace jsou vyobrazeny na obrázcích 4.6 a 4.7.



Obr. 4.6: Výsledky modelů (promíchaná data) – výkonnostní metriky



Obr. 4.7: Výsledky modelů (promíchaná data) – matice záměn

Shrnutí

V této podkapitole bylo vytvořeno několik řešení s různými přístupy, kde bylo užito 11 různých algoritmů strojového učení aplikovaných na označenou část datové sady HAI 22.04 (tedy soubory *test1.csv*, *test2.csv*, *test3.csv* a *test4.csv*). I bez složitějších úprav datové sady a použitých algoritmů bylo v nějakých případech dosaženo velice dobrých výsledků. Shrnutí všech přístupů je vyobrazeno v tab. 4.1. Tato tabulka vyobrazuje popis každého přístupu a algoritmus, který dosahoval v tomto přístupu nejlepších výsledků. Při vytváření zdrojových kódů pro strojové učení s učitelem bylo užito následujících zdrojů literatury: [80, 81, 82, 83, 85]. Všechny zdrojové kódy (sešity `Google Colab`) jsou součástí elektronické přílohy diplomové práce.

Tab. 4.1: Shrnutí výsledků strojového učení s učitelem.

Scénář	Výchozí řešení	Vyvážení trénovací datové sady	Změna poměrů datových sad	Ladění hyperparametrů	Odstranění datových bodů	Promíchání dat
Trénování	test4.csv	test4.csv	test1.csv test2.csv test4.csv	test4.csv	test4.csv	80 % z datových sad test1-test4.csv
Validace	20 % trénovací datové sady	20 % trénovací datové sady	20 % trénovací datové sady	20 % trénovací datové sady	20 % trénovací datové sady	-
Testování	test1.csv test2.csv test3.csv	test1.csv test2.csv test3.csv	test3.csv	test1.csv test2.csv test3.csv	test1.csv test2.csv test3.csv	20 % z datových sad test1-test4.csv
Nejlepší výsledek	Naive Bayes (test3.csv)	Naive Bayes (test3.csv)	Random Forest (test3.csv)	Naive Bayes (test3.csv)	Gradient Boosting (test3.csv)	Random Forest
Přesnost (Accuracy)	0,948	0,940	0,948	0,944	0,948	0,999
Preciznost (Precision)	0,673	0,528	0,871	0,588	0,798	0,998
Senzitivita (Recall)	0,298	0,307	0,190	0,303	0,207	0,981
F-míra (F-score)	0,413	0,388	0,312	0,400	0,329	0,990
FP	557	1051	108	817	201	4
FN	2696	2663	3111	2667	3046	45
TP	1145	1178	730	1164	795	2343
TN	58002	57508	58451	57742	58358	69848
Colab notebook	HAI_ML_1.ipynb	HAI_ML_2.ipynb	HAI_ML_3.ipynb	HAI_ML_4.ipynb	HAI_ML_5.ipynb	HAI_ML_6.ipynb

4.2.2 Neuronová síť

Neuronové sítě mohou být efektivní při detekci anomálií v úlohách binární klasifikace, zejména pokud se jedná o složité datové sady s mnoha vzorky dat. Neuronové sítě se mohou naučit relevantní vzorce v datech (např. změny hodnot v rámci datové sady HAI) a na základě toho určit, do jaké třídy data spadají ("0" pro legitimní data, a "1" pro anomálie).

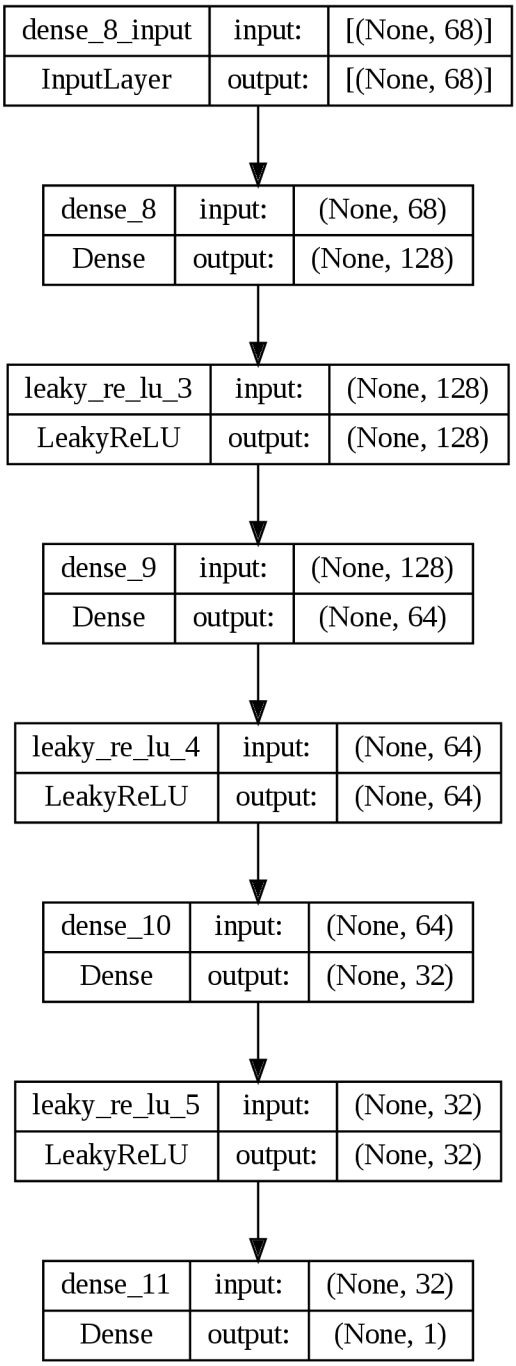
Vytvoření modelu neuronové sítě

Po patřičné úpravě datové sady (totožné úpravy, jak v minulé kapitole u strojového učení s učitelem, jen s tím rozdílem, že trénovací a validační datové sady jsou zamíchané) pro dosažení nejlepších možných výsledků proběhlo definování parametrů a hyperparametrů pro výchozí řešení před samotným trénováním modelu:

- **Datové sady:**
  - Trénovací data – 80 % test4.csv – 103 680 vzorků (3,41 % anomálií).
  - Validační data – 20 % test4.csv – 25 920 vzorků (3,43 % anomálií).
  - Testovací data:
    - test1.csv – 86 400 vzorků (1,03 % anomálií).
    - test2.csv – 82 800 vzorků (3,61 % anomálií).
    - test3.csv – 62 400 vzorků (6,16 % anomálií).
- **Počet epoch** – 200.
- **Velikost dávky** – 100.
- **Ztrátová funkce** – Binary Cross-entropy.
- **Optimalizátor** – Adam.

Při vytváření modelu neuronové sítě byla čerpána inspirace zejména ze zdroje literatury [86]. Bylo přistoupeno k řešení pomocí knihovny `TensorFlow`, konkrétně prostřednictvím rozhraní `Keras`. Modelování specifické neuronové sítě na problém detekce anomálií pomocí binární klasifikace představuje složitý problém, který vyžaduje velké množství experimentů pro dosažení optimálních výsledků na konkrétní řešení. Po prvotních experimentech byla sestavena neuronová síť následovně (model je vyobrazen na obr. 4.8):

- Plně propojená (dense) vrstva – 128 neuronů.
- Vrstva LeakyReLU.
- Plně propojená (dense) vrstva – 64 neuronů.
- Vrstva LeakyReLU.
- Plně propojená (dense) vrstva – 32 neuronů.
- Vrstva LeakyReLU.
- Výstupní vrstva s aktivační funkcí "sigmoid" – 1 neuron.



Obr. 4.8: Vyzualizace vytvořené neuronové sítě

Trénování, validace a testování modelu

Zdrojový kód pro vytvoření modelu neuronové sítě a následné trénování, validaci a testování tohoto modelu je vyobrazen v obr. 4.9.

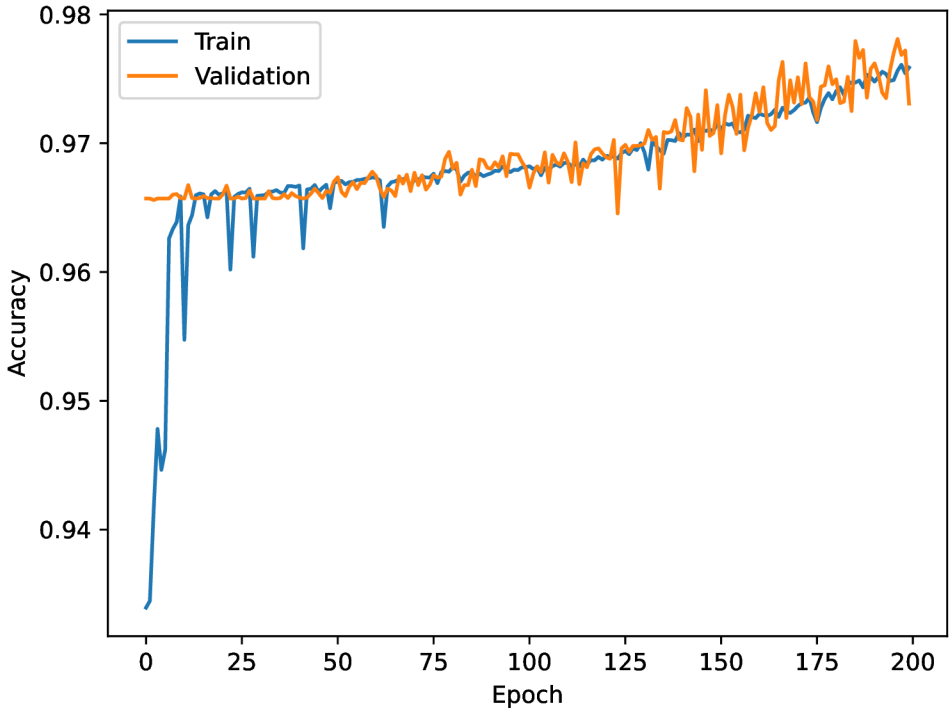
```
from keras.callbacks import EarlyStopping
from keras import regularizers

model = keras.Sequential([
    keras.layers.Dense(128, input_shape=(x_train.shape[1],)),
    keras.layers.LeakyReLU(alpha=0.1),
    keras.layers.Dense(64),
    keras.layers.LeakyReLU(alpha=0.1),
    keras.layers.Dense(32),
    keras.layers.LeakyReLU(alpha=0.1),
    keras.layers.Dense(1, activation='sigmoid')
])

# Kompilace modelu
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Definice funkce EarlyStopping
es = EarlyStopping(monitor='val_accuracy', mode='max', patience=20, restore_best_weights=True)

# Trénování modelu neuronové sítě
history = model.fit(x_train, y_train, epochs=200, batch_size=100,
                    validation_data=(x_valid, y_valid), callbacks=[es])

# Vyhodnocení modelu na testovacích datových sadách
test1_loss, test1_acc = model.evaluate(x_test1, y_test1, verbose=0)
test2_loss, test2_acc = model.evaluate(x_test2, y_test2, verbose=0)
test3_loss, test3_acc = model.evaluate(x_test3, y_test3, verbose=0)

# Predikce modelu
print('\nTesting:')
y_pred1 = np.round(model.predict(x_test1))
y_pred2 = np.round(model.predict(x_test2))
y_pred3 = np.round(model.predict(x_test3))

print('Test set 1 accuracy:', test1_acc, 'Test set 1 loss:', test1_loss)
print('Test set 2 accuracy:', test2_acc, 'Test set 2 loss:', test2_loss)
print('Test set 3 accuracy:', test3_acc, 'Test set 3 loss:', test3_loss)
```

Obr. 4.9: Definice modelu neuronové sítě – trénování, validace a testování

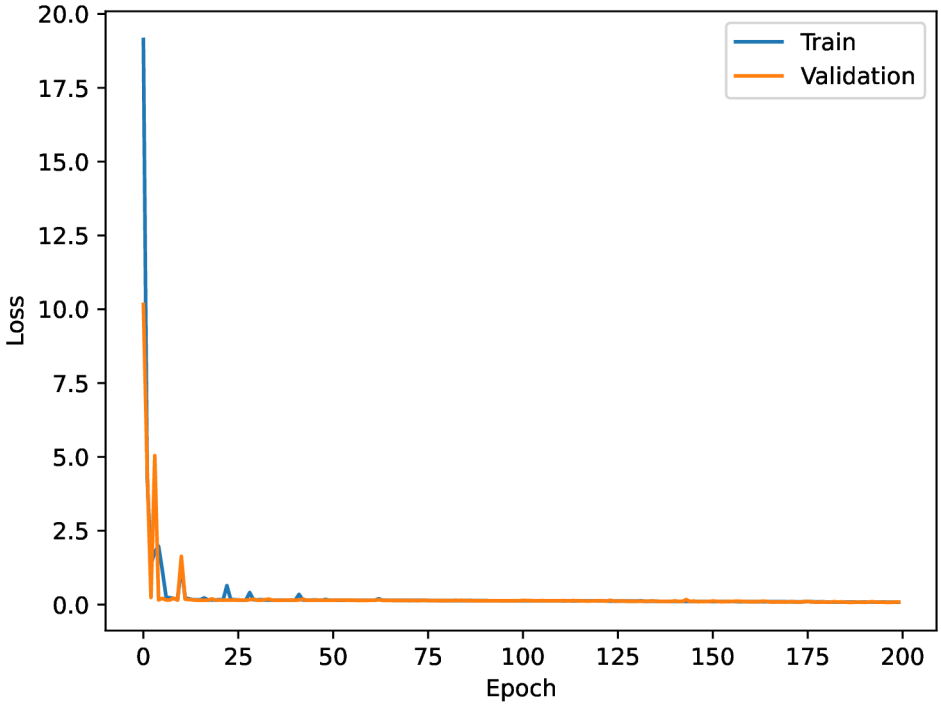
Po stanovení jednotlivých parametrů a hyperparametrů proběhlo trénování a validace modelu pomocí funkce `model.fit` s využitím funkce `EarlyStopping`, která umožňuje zastavení trénování a validace v případě, pokud se zvolená metrika (v tomto případě validační přesnost) nezlepšuje po určitý počet epoch (stanoveno na 20). V případě, že by nastala tato situace, budou obnoveny váhy modelu neuronové sítě z dané epochy, která vykazovala nejlepší výsledky. Výsledky trénování a validace modelu byly následující:

- **Ztráta při trénování** = 0,081.
- **Přesnost trénování** = 0,976.
- **Ztráta při validaci** = 0,087.
- **Přesnost validace** = 0,973.

Průběh přesnosti a ztráty při trénování a validaci modelu neuronové sítě je vykreslen na obr. 4.10 a obr. 4.11



Obr. 4.10: Přesnost trénování a validace modelu neuronové sítě (výchozí scénář)



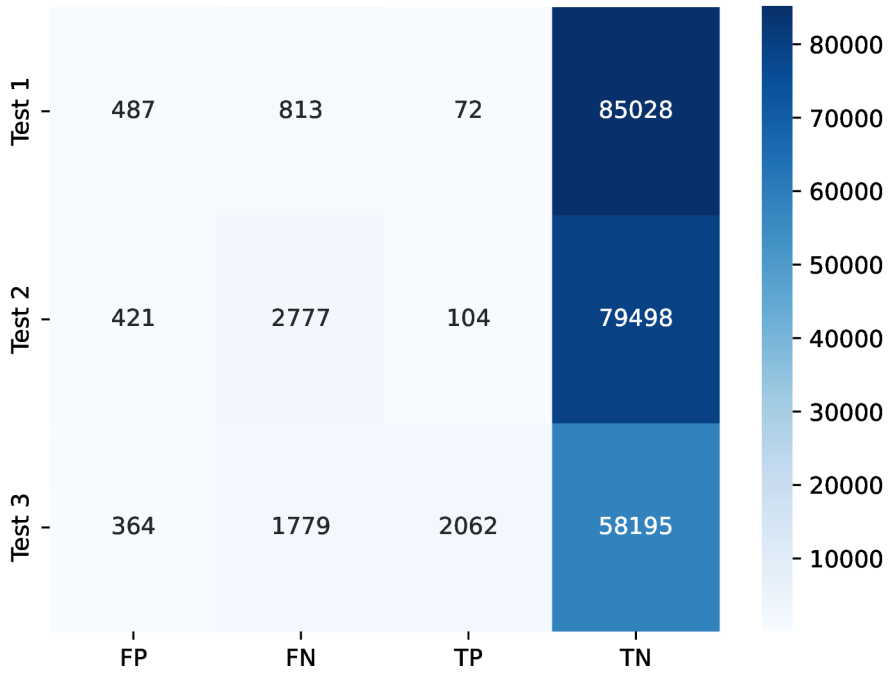
Obr. 4.11: Ztráta při trénování a validaci modelu neuronové sítě (výchozí scénář)

Výsledky modelu neuronové sítě

Z vyobrazených výsledků trénování a validace modelu neuronové sítě lze konstatovat, že dochází k soustavnému zlepšování (učení) modelu s mírnými výkyvy mezi jednotlivými epochami, což je nejspíše způsobeno užitím vrstev s aktivační funkcí LeakyReLU. Tento způsob chování modelu značí fakt, že tento proces neprobíhá plně optimálně. Vynecháním funkce LeakyReLU ovšem dochází k předejití situaci, kdy se jednotlivé neurony mohou přestat zlepšovat (učit). Výsledky při testování modelu na stanovených datových sadách byly následující:

- **test1.csv**:
  - Ztráta (loss) – 0,258.
  - Přesnost (accuracy) – 0,985.
  - Preciznost (precision) – 0,129.
  - Senzitivita (recall) – 0,081.
  - F–míra (f–score) –0,010.
- **test2.csv**:
  - Ztráta (loss) – 0,290.
  - Přesnost (accuracy) – 0,961.
  - Preciznost (precision) – 0,198.
  - Senzitivita (recall) – 0,036.
  - F–míra (f–score) –0,061.
- **test3.csv**:
  - Ztráta (loss) – 0,214.
  - Přesnost (accuracy) – 0,966.
  - Preciznost (precision) – 0,850.
  - Senzitivita (recall) – 0,537.
  - F–míra (f–score) – 0,658.

Ve výchozím řešení byla použita neupravená datová sada *test4.csv* bez standardizace, či normalizace. Z výsledků testování lze sledovat, že opět dochází k nepřesným predikcím anomálií v případě prvních dvou testovacích datových sad. V posledním případě (predikce anomálií na datové sadě *test3.csv*) byla korektně předpovězena více jak polovina anomálií s poměrně nízkou mírou falešně pozitivních predikcí, což svědčí o dobrém stavu neuronové sítě z hlediska predikcí na tuto datovou sadu. V případě prvních dvou souborů datových sad jsou predikce anomálií nepřesné (matice záměn je vyobrazena na obr. 4.12).



Obr. 4.12: Matice záměn modelu neuronové sítě (výchozí scénář)

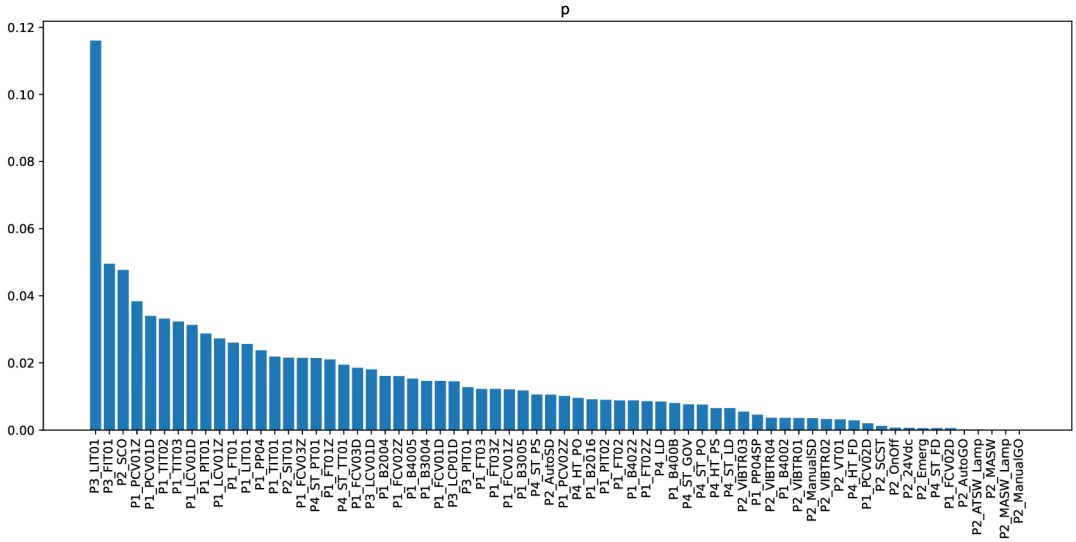
Optimalizace výsledků

Vzhledem k neoptimálním výsledkům výchozího scénáře lze spekulovat o přeučení modelu (overfitting). I přes postupné zlepšování přesnosti trénování i validace je možné, že došlo k přeučení modelu predikovat data jako legitimní, což je v tomto případu nežádoucí. Pokud se navíc bere v potaz mírná rozdílnost napříč jednotlivými datovými sadami, výsledkem jsou nedostatečně optimální predikce modelu. Pro optimalizaci výsledků byly provedeny následující scénáře:

- **Vyvážení trénovací datové sady** – byl vyzkoušen scénář s vyvážením datové sady pomocí funkce RandomOverSampler, což ovšem vedlo pouze ke zvýšení míry FP a k žádnému zlepšení na straně korektních predikcí anomálií TP. V této situaci proběhlo opět vyvážení trénovací datové sady náhodným doplněním anomálních vzorků dat, a ve výsledku tedy vznikla datová sada se 125 177 vzorky od každé třídy. Špatné výsledky model poskytoval z důvodu přeučení na anomální vzorky a tím pádem je i více predikoval na testovacích datových sadách. Ve funkci RandomOverSampler byla provedena úprava – přidání parametru sampling_strategy=1: 20000. Tento parametr zajistil, aby se anomální vzorky doplňovaly pouze do počtu 20 000. Tato úprava značně zlepšila predikce TP, ovšem pouze na datové sadě *test3.csv*, přičemž opět došlo ke zvýšení predikcí FP oproti výchozímu řešení.

- **Změna poměrů datových sad** – v tomto případě bylo přistoupeno k řešení, kde pro učení byly užity datové sady *test1.csv*, *test2.csv* a *test4.csv*, a k testování byla použita datová sada *test3.csv*. Tento scénář byl vyzkoušen nejdříve s náhodně promíchanými daty, a posléze bez promíchání. Nejlepšího výsledku bylo dosaženo po 100 epochách. V porovnání s výchozím řešením lze pozorovat nižší míru FP, ale rovněž i nižší TP, z čehož vyplývá vysoká preciznost, ale nižší senzitivita a f–míra.
- **Odstranění redundantních datových bodů** – totožná situace, jako u tohoto optimalizačního kroku u strojového učení s učitelem (analýza důležitostí datových bodů z obrázků B.1, B.2, B.3 a B.4). Tentokrát bylo přistoupeno pouze k řešení, které v rámci strojového učení s učitelem dosahovalo nejlepších výsledků, tedy bylo ve všech datových sadách ponecháno pouze 17 datových bodů z výchozích 68. Výsledky tohoto scénáře jsou horší, než ve výchozím scénáři. Neuronová síť má problém se efektivně naučit souvislosti z tak malého množství datových bodů.
- **Feature importance** – v tomto řešení je opět provedeno odstranění redundantních datových bodů z datové sady, pouze s tím rozdílem, že je k určení důležitosti jednotlivých datových bodů použit klasifikátor RandomForest. Pomocí něj je vyhodnocena důležitost všech datových bodů oproti cílové proměnné (tedy zda se jendá o legitimní provoz, či anomálii). S pomocí této metody byly datové sady redukovány na 33 datových bodů. V tomto případě bylo dosaženo značně lepších výsledků, než v posledním scénáři, ale horších, než ve scénáři "Změna poměrů datových sad". Vyobrazení důležitosti jednotlivých datových bodů je možné sledovat na obr. 4.13.

Obr. 4.13: Feature importance jednotlivých datových bodů.



- **Promíchání datových sad** – v tomto scénáři byly datové sady spojeny dohromady a náhodně rozděleny v poměru 48:20:32 (trénovací:validační:testovací datová sada). Pro vytvoření těchto tří datových sad je užito train_test_split funkce dvakrát po sobě. Poprvé na rozdělení originální datové sady (spojené soubory *test1.csv*, *test2.csv*, *test3.csv* a *test4.csv*) v poměru 80:20, a podruhé při rozdělení vytvořené trénovací datové sady (80 % z originálu) na trénovací a testovací v poměru 60:40. V tomto řešení byly také přidány další vrstvy neuronové sítě, konkrétně vrstva BatchNormalization pro optimalizaci procesu učení, a vrstva Droupout pro předejití přetrénování modelu (overfitting). Také byla přidána další plně propojená vrstva se 16 neurony. Byla v tomto případě také použita datová sada se 33 datovými body (po provedení feature importance). Výsledky byly lepší s ohledem na preciznost modelu, tedy nižší míry FP, ovšem míra korektně určených TP byla nižší, než ve výchozím řešení.

Shrnutí

Tato podkapitola zahrnuje modelování neuronové sítě, validaci prvotních výsledků, a úprava modelu neuronové sítě do fáze, kdy dává aspoň z části optimální výsledky pro tvorbu výchozího scénáře. Dále se podkapitola zabývá různými scénáři a technikami používanými k optimalizaci výkonu neuronové sítě při detekci anomálií. Patří mezi ně vyvážení trénovací datové sady, změna poměrů datových sad, odstranění nadbytečných datových bodů a provedení analýzy důležitosti datových bodů (feature importance). Některé přístupy vykazují zlepšení pravdivě pozitivních predikcí (TP), zatímco jiné vedou k lepší preciznosti, ale nižší citlivosti a F-skóre. Nejlepších výsledků je dosaženo redukcí datové sady na 33 nejvíce důležitých datových bodech pomocí metody feature importance. Všechny výsledky jsou shrnuty v tab. 4.1. V tabulce je možné sledovat nejlepší dosažené výsledky výchozího scénáře a všech optimalizačních řešení. Parametr "počet epoch"v tabulce značí, u které epochy bylo trénování zastavenou přičiněním funkce EarlyStopping. Jak již bylo zmíněno, maximální počet epoch byl stanoven na 200. Scénáře zmíněné v tabulce byly vyzkoušeny i se standardizací, a normalizací datových sad (pomocí StandardScaler, a MinMaxScaler), což ovšem nevedlo ke zlepšení výsledků, nýbrž ke skoro stejným, či horším výsledkům. Všechny zdrojové kódy (sešity `Google Colab`) jsou součástí elektronické přílohy diplomové práce.

Tab. 4.2: Shrnutí výsledků neuronových sítí.

Scénář	Výchozí řešení	Vyvážení trénovací datové sady	Změna poměrů datových sad	Odstranění datových bodů	Feature importance	Promíchání dat
Trénování	test4.csv	test4.csv	test1.csv test2.csv test4.csv	test1.csv test2.csv test4.csv	test1.csv test2.csv test4.csv	48 % z datových sad test1-test4.csv
Validace	20 % trénovací datové sady	20 % trénovací datové sady	20 % trénovací datové sady	20 % trénovací datové sady	20 % trénovací datové sady	20 % z datových sad test1-test4.csv
Testování	test1.csv test2.csv test3.csv	test1.csv test2.csv test3.csv	test3.csv	test3.csv	test3.csv	32 % z datových sad test1-test4.csv
Model	Výchozí	Výchozí	Výchozí	Výchozí	Výchozí	Dropout, Batch Normalization
Počet epoch	200	94	100	200	200	136
Trénovací přesnost	0,976	0,959	0,979	0,937	0,978	0,978
Trénovací ztráta (loss)	0,081	0,105	0,082	0,264	0,083	0,085
Validační přesnost	0,973	0,977	0,970	0,929	0,979	0,980
Validační ztráta (loss)	0,087	0,061	0,157	0,235	0,091	0,082
Přesnost (Accuracy)	0,966	0,932	0,964	0,943	0,971	0,979
Preciznost (Precision)	0,850	0,459	0,976	0,600	0,938	0,937
Senzitivita (Recall)	0,537	0,551	0,433	0,248	0,564	0,390
F-míra (F-score)	0,658	0,501	0,600	0,351	0,705	0,551
FP	364	2492	41	645	142	101
FN	1779	1724	2179	2887	1675	2345
TP	2062	2117	1662	954	2166	1502
TN	58195	56067	58518	57914	58417	111636
Colab notebook	HAI_NN_1.ipynb	HAI_NN_2.ipynb	HAI_NN_3.ipynb	HAI_NN_4.ipynb	HAI_NN_5.ipynb	HAI_NN_6.ipynb

4.2.3 Strojové učení bez učitele

V poslední řadě bylo přistoupeno k otestování algoritmů strojového učení bez učitele pro zjištění, jak dobře budou schopny predikovat anomálie v porovnání s algoritmy strojového učení s učitelem, či neuronovými sítěmi. Strojové učení bez učitele analyzuje data bez označení a snaží se v nich najít souvislosti. Z tohoto důvodu jsou tyto algoritmy hojně využívány pro hledání takových dat, která se liší od ostatních (tedy anomálií). V rámci datové sady HAI je ovšem tento úkol problematický, protože, jak již bylo zmíněno v analýze důležitosti datových bodů, anomálie jsou v datech rozprostřeny takovým způsobem, že ne vždy nabývají jiných hodnot oproti legitimním datům. Z tohoto důvodu byly v rámci tohoto řešení otestovány pouze 3 algoritmy strojového učení bez učitele – konkrétně Isolation Forest, One-Class SVM, a LOF (*Local Outlier Factor*).

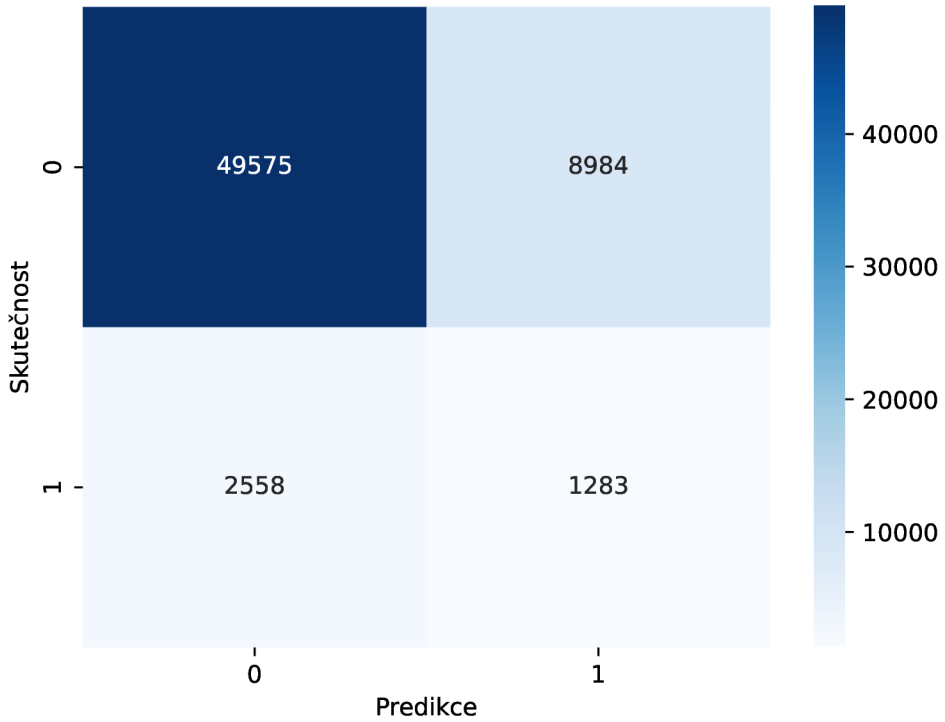
Scénáře pro testování zvolených algoritmů strojového učení bez učitele byly zhotoveny následovně:

- **Trénovací datová sada** – test4.csv.
- **Testovací datová sada** – test3.csv.
- **Úprava dat** – přístup feature importance (užito 33 nejrelevantnějších datových bodů).

V tomto přístupu nebylo využito validačních dat, jelikož bez označení jednotlivých prvků neexistuje způsob, jak by se dala ověřit korektnost prvotních predikcí algoritmů na neznámých datech. Po stanovení všech nutných parametrů pro testování jednotlivých algoritmů proběhlo vyhotovení zdrojového kódu v sešitu `Google Colab`. Provedením počátečních testů bylo přistoupeno k závěru, že algoritmy strojového učení bez učitele vykazují velice neoptimální výsledky, a nejsou tedy pro datovou sadu HAI plně vhodné. Následuje výpis nejlepších výsledků algoritmu One-Class SVM (matice záměn je vyobrazena na obr. 4.14):

- **Přesnost (Accuracy)** – 0,815.
- **Preciznost (Precision)** – 0,125.
- **Senzitivita (Recall)** – 0,334.
- **F–míra (F–score)** – 0,182.
- **FP** – 8 984; **FN** – 2 558; **TP** – 1 283; **TN** – 49 575.

Obr. 4.14: Testování algoritmu One-Class SVM – matice záměn.



5 Výsledky práce

V rámci této kapitoly jsou diskutovány důležité skutečnosti ohledně realizace praktické části diplomové práce a jejich výsledků.

Shrnutí praktické části

V praktické části byla provedena podrobná analýza detekce anomálií s využitím modelů strojového učení s učitelem a neuronových sítí. V rámci těchto dvou přístupů byl vyhotoven podrobný popis všech užitých algoritmů a také popis potřebného zpracování datové sady pro možnost vyhotovení těchto přístupů. V rámci strojového učení s učitelem bylo užito 11 různých algoritmů, kde nejlepších výsledků dosahovaly algoritmy Random Forest a Naive Bayes. S pomocí úprav a optimalizace datové sady byly dosaženy velice různorodé výsledky, což dokazuje, jak markantní je optimalizace datové sady před jejím užití pro trénování, validaci a testování modelů.

V rámci strojového učení s učitelem bylo dosaženo nejlepších výsledků po promíchání datových sad, bez jejich standardizace, či normalizace. V tomto případě z testovaných 11 algoritmů dosahoval nejlepších výsledků Random Forest s následujícími výkonnostními metrikami:

- **Přesnost (Accuracy)** – 0,999.
- **Preciznost (Precision)** – 0,998.
- **Senzitivita (Recall)** – 0,981.
- **F–míra (F–score)** – 0,990.
- **FP** – 4; **FN** – 45; **TP** – 2 343; **TN** – 69 848.

V případě neuronových sítí bylo provedeno experimentování s různými modely (změnami samotných vrstev, počtu neuronů, aktivačních funkcí atd.), po čemž mohlo nastat vytvoření finálního modelu, který dosahoval pro výchozí scénář nejlepší výsledky. Varianta vykazující nejlepší řešení vznikla provedením analýzy feature importance na datové sadě, přičemž bylo určeno 33 nejdůležitějších datových bodů z celkových 68. Výsledky tohoto řešení byly následující po 200 epochách:

- **Trénovací přesnost** – 0,978.
- **Trénovací ztráta** – 0,083.
- **Validační přesnost** – 0,979.
- **Validační ztráta** – 0,091.
- **Přesnost (Accuracy)** – 0,971.
- **Preciznost (Precision)** – 0,938.
- **Senzitivita (Recall)** – 0,564.
- **F–míra (F–score)** – 0,705.
- **FP** – 142; **FN** – 1 675; **TP** – 2 166; **TN** – 58 417.

V případě posledního řešení, tedy vyhotovení scénáře pro strojové učení bez učitele, bylo přistoupeno k implementaci pouze jednoho algoritmu, konkrétně One-class SVM. Využité datové sady byly opět redukovány na 33 nejdůležitějších datových bodů. Zprvu byly vyzkoušeny další dva algoritmy – Isolation Forest a LOF. Při prvotních testech nebylo dosaženo optimálních výsledků ani v jednom ze všech tří algoritmů. I přes různé modifikace, jak užitých algoritmů, tak i datových sad, byly nejlepší dosažené výsledky stále velice neoptimální (algoritmus One-class SVM):

- **Přesnost (Accuracy)** – 0,815.
- **Preciznost (Precision)** – 0,125.
- **Senzitivita (Recall)** – 0,334.
- **F–míra (F–score)** – 0,182.
- **FP** – 8 984; **FN** – 2 558; **TP** – 1 283; **TN** – 49 575.

Důležité poznatky

V rámci zpracování jednotlivých řešení byl kladen důraz na podrobný popis zpracování datové sady. V rámci projednávaných vědeckých článků (jak v provedené rešerši, tak i v řešeních pracujících s datovou sadou HAI) není této problematice věnován dostatečný prostor. V [90] je např. podobně rozepsána, mimo jiné, datová sada HAI, ale nejsou specifikovány informace o poměrech a specifických souborech datových sad. Tato informace představuje velice důležitou část celého procesu aplikace strojového učení na datovou sadu HAI. Jak je zmíněno v předchozí kapitole, jakákoliv úprava datové sady má nemalý vliv na konečné výsledky jednotlivých modelů.

Přínos této diplomové práce spočívá v podrobném popisu práce s daty, kde je realizováno mnoho různých úprav datových sad a užití velkého rozsahu algoritmů. Odvedená práce se tedy dá použít jako inspirace, či jako podklad pro budoucí výzkum v této oblasti. Problematika detekce anomálií v ICS pomocí strojového učení je ovšem problematická z důvodu velké specifičnosti různých dostupných datových sad. Tento problém se vztahuje i na samotné datové sady, jelikož v případě HAI existuje značná rozdílnost hodnot datových bodů v rámci souborů *test1.csv*, *test2.csv*, *test3.csv* a , což je v tomto případě pravděpodobná příčina neoptimálních výsledků strojového učení bez učitele. Tato rozdílnost komplikuje aplikaci jakéhokoliv algoritmu, jelikož naučená data jsou z tohoto důvodu nepřesná – parametry jednotlivých prvků, pokud se jedná o anomálii, se mohou napříč datovými sadami lišit. I přes tahle negativa by se na základě této práce dalo navrhnout např. další model neuronové sítě, a opět provést celý proces trénování, validace, testování, spolu se všemi optimalizačními kroky pro možný rozvoj a zdokonalení výsledků detekce anomálií. V tomto případě lze pracovat i s vytvořenými sešity Google Colab, které jsou součástí elektronických příloh (zmíněno v kapitole C).

Závěr

V práci byla provedena analýza oblasti průmyslových řídících systémů z hlediska bezpečnosti. V teoretické části práce byly zmíněny základní informace o průmyslových řídících systémech, jejich fungování, nedostatcích a možných vylepšeních v rámci jejich zabezpečení. Následně byly popsány různé typy anomálií a narušení, a jejich příčiny. Na základě tohoto výzkumu byly zpracovány možné metody detekce anomálií a narušení a byl shrnut současný stav oblasti. V rámci možnosti detekce anomálií byla zpracována rešerše na současně dostupné datové sady pro průmysl, které se dají použít právě pro trénování modelů strojového učení pro možné zlepšení současného stavu průmyslových řídících systémů. Byl také proveden podrobný popis zvolené datové sady (HAI 22.04), s pomocí které lze pracovat na účinném vylepšení detekce anomálií díky jejímu dobrému popisu z hlediska dokumentace. Před samotným aplikováním strojového učení na zvolenou datovou sadu byly rozebrány základy umělé inteligence v rámci oblasti strojového učení a neuronových sítí.

V rámci praktické části diplomové práce proběhlo stanovení strategie k dosažení detekce anomálií na datové sadě HAI. K tomuto úkolu byla vybrána binární klasifikace pomocí základních algoritmů – strojového učení s učitelem a bez učitele, a pokročilejších algoritmů – neuronové sítě. Ke zpracování zdrojového kódu bylo využito vývojové prostředí Google Colab. Pro účel praktické části diplomové práce byla zpracována pouze část datové sady, kde jsou data označena. Bylo provedeno trénování, validace a testování různých modelů jednotlivých přístupů a proběhlo zaznamenání všech důležitých poznatků. Na základě těchto poznatků bylo následně realizováno několik scénářů pro optimalizaci dosažených výsledků v rámci strojového učení s učitelem a neuronových sítí. Nakonec byly výsledky jednotlivých řešení znázorněny, a byly vybrány ty nejlepší z nich.

V poslední kapitole práce jsou okomentovány a shrnuty nejlepší výsledky všech řešení. Také jsou zde zahrnuty důležité poznatky o problematice detekce anomálií v ICS. Je také zmíněn návrh na budoucí rozvoj v této oblasti, a možné navázání na tuto diplomovou práci.

Literatura

[1] HU, Y.; YANG, A.; LI, H.; SUN, Y.; SUN, L.: *A survey of intrusion detection on industrial control systems*. [online]. [cit. 25. 9. 2022]. Dostupné z URL: <https://journals.sagepub.com/doi/full/10.1177/1550147718794615>.

[2] SHABTAI, A.; MEIDAN, Y.; KRAVCHIK, M.; GURION, B.: *Anomaly detection for industrial control systems*. [online]. [cit. 25. 9. 2022]. Dostupné z URL: <https://www.concordia-h2020.eu/blog-post/anomaly-detection-for-industrial-control-systems-ics/>.

[3] MOKHTARI, S.; ABBASPOUR, A.; KANG, K. Y.; SARGOLZAEI, A.: *A Machine Learning Approach for Anomaly Detection in Industrial Control Systems Based on Measurement Data*. [online]. [cit. 25. 9. 2022]. Dostupné z URL: <https://www.mdpi.com/2079-9292/10/4/407/htm>.

[4] RealPars: *Automation Pyramid?*. [online]. [cit. 25. 9. 2022]. Dostupné z URL: <https://realpars.com/automation-pyramid/>.

[5] Trend Micro: *Industrial Control System*. [online]. [cit. 25. 9. 2022]. Dostupné z URL: <https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system>.

[6] MATHEZER, S.: *Introduction to ICS security fundamentals*. [online]. [cit. 25. 9. 2022]. Dostupné z URL: <https://www.industrialcybersecuritypulse.com/education/introduction-to-ics-security-fundamentals/>.

[7] KUMAR, R.: *What is the five layer automation pyramid?*. [online]. [cit. 10. 10. 2022]. Dostupné z URL: <https://medium.com/world-of-iot/92-what-is-the-five-layer-automation-pyramid-d0ccc1b903c3>.

[8] POSPÍŠIL, O.; BLAŽEK, P.; KUCHAŘ, K.; FUJDIAK, R.; MIŠUREC, J.: *Application Perspective on Cybersecurity Testbed for Industrial Control Systems*. [online]. [cit. 10. 10. 2022]. Dostupné z URL: <https://www.mdpi.com/1424-8220/21/23/8119/htm>.

[9] MILLER, T.; STAVES, A.; MAESSCHALCK, S.; STURDEE., M.; GREEN, B.: *Looking Back to Look Forward: Lessons Learnt from Cyber-Attacks on Industrial Control Systems*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://www.sciencedirect.com/science/article/abs/pii/S1874548221000524>.

[10] IRMAK, E.; ERKEK, I.: *An Overview of Cyber-Attack Vectors on SCADA Systems*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://ieeexplore.ieee.org/abstract/document/8355379>.

[11] ANTHI, E.; WILLIAMS, L.; RHODE, M.; BURNAP, P.; WEDGBURY, A.: *Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://www.sciencedirect.com/science/article/pii/S2214212620308607>.

[12] CHIOCK, M.; RODILLAS, D.: *Kybernetická bezpečnost průmyslových řídicích systémů (část 1)*. [online]. [cit. 26. 9. 2022]. Dostupné z URL: <http://automa.cz/Aton/FileRepository/pdf_articles/54542.pdf>.

[13] RIBEROLLES, T. de; ZOU, Y.; SILVESTRE, G.; LOCHIN, E.; SONG, J.: *Anomaly detection for ICS based on deep learning: a use case for aeronautical radar data*. [online]. [cit. 30. 9. 2022]. Dostupné z URL: <https://link.springer.com/content/pdf/10.1007/s12243-021-00902-7.pdf>.

[14] BHAMARE, D.; ZOLANVARI, M.; ERBAD, A.; JAIN, R.; KHAN, K.; MESKIN, N.: *Cybersecurity for industrial control systems: A survey*. [online]. [cit. 6. 10. 2022]. Dostupné z URL: <https://www.sciencedirect.com/science/article/pii/S0167404819302172>.

[15] CHANDOLA, V.; BANERJEE, A.; KUMAR, V.: *Anomaly detection: A survey*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://dl.acm.org/doi/abs/10.1145/1541880.1541882>.

[16] KAMPAKIS, S.: *3 Types of Anomalies in Anomaly Detection*. [online]. [cit. 20. 10. 2022]. Dostupné z URL: <https://hackernoon.com/3-types-of-anomalies-in-anomaly-detection>.

[17] HA, T. D.; HOANG, X. N.; DU, H N.; HOANG, V. N.; HUONG, T. T.; TRAN, P. K.: *Explainable Anomaly Detection for Industrial Control System Cybersecurity*. [online]. [cit. 20. 10. 2022]. Dostupné z URL: <https://arxiv.org/pdf/2205.01930.pdf>.

[18] GOH, J.; ADEPU, S.; JUNEJO, K. N.; MATHUR, A.: *A Dataset to Support Research in the Design of Secure Water Treatment Systems*. [online]. [cit. 20. 10. 2022]. Dostupné z URL: <https://www.researchgate.net/profile/Khurum-Junejo/publication/305809559_A_Dataset_to_Support_Research_in_the_Design_of_Secure_Water_Treatment_Systems/links/57a2beb308ae5f8b258cb437/A-Dataset-to-Support-Research-in-the-Design-of-Secure-Water-Treatment-Systems.pdf>.

[19] ASHFORD, W.: *Human error a big risk to ICS cyber security, study shows*. [online]. [cit. 24. 10. 2022]. Dostupné z URL: <https://www.computerweekly.com/news/252468880/Human-error-a-big-risk-to-ICS-cyber-security-study-shows>.

[20] MICHAEL, M.: *3 Challenges in Securing Industrial Control Systems*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://blog.f-secure.com/3-challenges-in-securing-industrial-control-systems-and-3-solutions/>.

[21] ZABEU, S.: *More than 70 % of failures in industrial control systems are critical*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://network-king.net/more-than-70-of-failures-in-industrial-control-systems-are-critical/>.

[22] Vedere Labs: *OT:ICEFALL: 56 Vulnerabilities Caused by Insecure-by-Design Practices in OT*. [online]. [cit. 23. 11. 2022]. Dostupné z URL: <https://www.forescout.com/blog/ot-icefall-56-vulnerabilities-caused-by-insecure-by-design-practices-in-ot/>.

[23] DEWA, Z.; MAGLARAS, L.: *Data Mining and Intrusion Detection Systems*. [online]. [cit. 24. 10. 2022]. Dostupné z URL: <https://www.researchgate.net/publication/289957493_Data_Mining_and_Intrusion_Detection_Systems>.

[24] JOSHI, A.; FININ, T.: *A Knowledge-Based Approach To Intrusion Detection Modeling*. [online]. [cit. 24. 10. 2022]. Dostupné z URL: <https://www.researchgate.net/publication/261488988_A_Knowledge-Based_Approach_to_Intrusion_Detection_Modeling>.

[25] GAVRILOVA, Y.: *What Is Anomaly Detection in Machine Learning?*. [online]. [cit. 24. 10. 2022]. Dostupné z URL: <https://serokell.io/blog/anomaly-detection-in-machine-learning>.

[26] FALLIERE, N.; MURCHU, L. O; CHIEN, E.: *W32.Stuxnet Dossier*. [online]. [cit. 27. 10. 2022]. Dostupné z URL: <https://pax0r.com/hh/stuxnet/Symantec-Stuxnet-Update-Feb-2011.pdf>.

[27] BOYES, H.; HALLAQ, B.; CUNNINGHAM, J.; WATSON, T.: *The industrial internet of things (IIoT): An analysis framework*. [online]. [cit. 26. 10. 2022]. Dostupné z URL: <https://www.sciencedirect.com/science/article/pii/S0166361517307285>.

[28] WANG, CH.; WANG, B.; LIU, H.; QU, H.: *Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network*. [online]. [cit. 26. 10. 2022]. Dostupné z URL: <https://www.researchgate.net/publication/343424305_Anomaly_Detection_for_Industrial_Control_System_Based_on_Autoencoder_Neural_Network>.

[29] BLÜMELOVÁ, K. K.: *Více než 1 600 firem nechává „otevřené dveře“ do průmyslových řídicích systémů pro nezvané hosty!*. [online]. [cit. 7. 11. 2022]. Dostupné z URL: <https://www.technickytydenik.cz/rubriky/ict/vice-nez-1-600-firem-nechava-otevrene-dvere-do-prumyslovych-ridicich-systemu-pro-nezvane-hosty_56071.html>.

[30] KOAY, A. M. Y.; KO, R. K. L; HETTEMA, H.; RADKE, K.: *Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges*. [online]. [cit. 8. 11. 2022]. Dostupné z URL: <https://link.springer.com/article/10.1007/s10844-022-00753-1>.

[31] Asociace kritické infrastruktury České republiky: *Kritická infrastruktura*. [online]. [cit. 9. 11. 2022]. Dostupné z URL: <https://www.akicr.cz/kriticka-infrastruktura/>.

[32] de RIBEROLLES, T.; ZOU, Y.; SILVESTRE, G.; LOCHIN, E; SONG, J.: *Anomaly detection for ICS based on deep learning: a use case for aeronautical radar data*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://link.springer.com/article/10.1007/s12243-021-00902-7.

[33] FENG, C.; LI, T.; CHANA, D.: *Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM networks*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://orca.cardiff.ac.uk/id/eprint/127039/1/Li_DSN17.pdf.

[34] KRAVCHIK, M.; SHABTAI, A.: *Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://dl.acm.org/doi/10.1145/3264888.3264896.

[35] WANG, CH.; WANG, B.; LIU, H.; QU, H.: *Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://www.hindawi.com/journals/wcmc/2020/8897926/.

[36] VÁVRA, J.; HROMADA, M.; LUKÁŠ, L.; DWORZECKI, J.: *Adaptive anomaly detection system based on machine learning algorithms in an industrial*

control environment. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://www.sciencedirect.com/science/article/pii/S187454822100038X.

[37] INOUE, J.; YAMAGATA, Y.; CHEN, Y.; POSKITT, M. CH.; SUN, J.: *Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://arxiv.org/pdf/1709.05342.pdf.

[38] AL-HAWAWREH, M.; MOUSTAFA, N.; SITNIKOVA, E.: *Identification of malicious activities in industrial internet of things based on deep learning models*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://www.sciencedirect.com/science/article/pii/S2214212617306002.

[39] AL-ABASSI, A.; KARIMIPOUR, H.; DEHGHANTANHA, A.; PARIZI, M. R.: *An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9086038.

[40] DAS, K. T.; ADEPU, S.; ZHOJ, J.: *Anomaly detection in Industrial Control Systems using Logical Analysis of Data*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://www.sciencedirect.com/science/article/pii/S0167404820302121.

[41] WANG, W.; WANG, Z.; ZHOJ, Z.; DENG, H.; ZHAO, W.; WANG, CH.; GUO, Y.: *Anomaly detection of industrial control systems based on transfer learning*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9449327.

[42] SHALYGA, D.; FILONOV, P.; LAVRENTYEV, A.: *Anomaly Detection for Water Treatment System based on Neural Network with Automatic Architecture Optimization*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://arxiv.org/pdf/1807.07282.pdf.

[43] LI, D.; CHEN, D.; SHI, L.; JIN, B.; GOH, J.; NG, S.-K.: *MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://arxiv.org/pdf/1901.04997.pdf.

[44] KIM, J.; YUN, J.-H.; KIM, H. CH.: *Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://arxiv.org/pdf/1911.04831.pdf.

[45] KRAVCHIK, M.; SHABTAI, A.: *Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://arxiv.org/pdf/1907.01216.pdf.

[46] LIU, L.; HU, M.; KANG, CH.; LI, X.: *Unsupervised Anomaly Detection for Network Data Streams in Industrial Control Systems*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://www.mdpi.com/2078-2489/11/2/105/htm.

[47] SCHNEIDER, P.; BOTTINGER, K.: *High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://dl.acm.org/doi/pdf/10.1145/3264888.3264890.

[48] ELNOUR, M.; MESKIN, N.; KHAN, K.; JAIN, R.: *A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9003235.

[49] KHAN, A. AL Z.; SERPEN, G.: *Misuse Intrusion Detection Using Machine Learning for Gas Pipeline SCADA Networks*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://www.researchgate.net/publication/333248611.

[50] ALHAIDARI, F. A.; AL-DAHASI, E. M.: *New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://ieeexplore.ieee.org/abstract/document/8716432.

[51] GOMEZ, A. L. P.; MAIMO, L. F.; CELDRAN, A. H.; CLEMENTE, F. J. G.; SARMIENTO, C. C.; MASA, C. J. del C.; NISTAL, R. M.: *On the Generation of Anomaly Detection Datasets in Industrial Control Systems*. [online]. [cit. 26. 3. 2022]. Dostupné z URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8926471.

[52] KAVLAKOGLU, E.: *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?*. [online]. [cit. 20. 11. 2022]. Dostupné z URL: <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>.

[53] SELIG, J.: *What Is Machine Learning? A Definition*. [online]. [cit. 20. 11. 2022]. Dostupné z URL: <https://www.expert.ai/blog/machine-learning-definition/>.

[54] GeeksforGeeks: *Machine Learning*. [online]. [cit. 20. 11. 2022]. Dostupné z URL: <https://www.geeksforgeeks.org/machine-learning/>.

[55] CHEN, J.: *What Is a Neural Network?* [online]. [cit. 20. 11. 2022]. Dostupné z URL: <https://www.investopedia.com/terms/n/neuralnetwork.asp>.

[56] IBM Cloud Education: *Neural Networks.* [online]. [cit. 20. 11. 2022]. Dostupné z URL: <https://www.ibm.com/cloud/learn/neural-networks>.

[57] DOBILAS, S.: *Feed Forward Neural Networks – How To Successfully Build Them in Python.* [online]. [cit. 20. 11. 2022]. Dostupné z URL: <https://towardsdatascience.com/feed-forward-neural-networks-how-to-successfully-build-them-in-python-74503409d99a>.

[58] The Affiliated Institute of ETRI, South Korea: *HIL-based Augmented ICS Security Dataset.* [online]. [cit. 10. 11. 2022]. Dostupné z URL: <https://github.com/icsdataset/hai>.

[59] *What Is Hardware-in-the-Loop?* [online]. [cit. 10. 11. 2022]. Dostupné z URL: <https://www.ni.com/cs-cz/solutions/transportation/hardware-in-the-loop/what-is-hardware-in-the-loop-.html>.

[60] BAE, S.; HWANG, CH.; LEE, T.: *Research on Improvement of Anomaly Detection Performance in Industrial Control Systems.* [online]. [cit. 25. 11. 2022]. Dostupné z URL: <https://link.springer.com/chapter/10.1007/978-3-030-89432-0_7>.

[61] MOKHTARI, S.; YEN, K. K.; HUNG, C.–C.: *Measurement data intrusion detection in industrial control systems based on unsupervised learning.* [online]. [cit. 25. 11. 2022]. Dostupné z URL: <https://www.aimspress.com/article/doi/10.3934/aci.2021004>.

[62] KUMAR, A.; CHOI, B. J.: *Benchmarking Machine Learning based Detection of Cyber Attacks for Critical Infrastructure.* [online]. [cit. 25. 11. 2022]. Dostupné z URL: <https://ieeexplore.ieee.org/abstract/document/9687293>.

[63] FF12, Cloudera: *Deep Learning for Anomaly Detection.* [online]. [cit. 7. 12. 2022]. Dostupné z URL: <https://ff12.fastforwardlabs.com/>.

[64] FUMO, D.: *Types of Machine Learning Algorithms You Should Know.* [online]. [cit. 7. 12. 2022]. Dostupné z URL: <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>.

[65] RAMAKRISHNAN, M.: *Types of Machine Learning.* [online]. [cit. 7. 12. 2022]. Dostupné z URL: <https://emeritus.org/blog/types-of-machine-learning/>.

[66] BEKLEMYSHEVA, A.: *Why Use Python for AI and Machine Learning?*. [online]. [cit. 8. 12. 2022]. Dostupné z URL: <https://steelkiwi.com/blog/python-for-ai-and-machine-learning/>.

[67] DECLAN, V.: *Python Data Analysis with Pandas and Matplotlib*. [online]. [cit. 25. 4. 2023]. Dostupné z URL: <https://ourcodingclub.github.io/tutorials/pandas-python-intro/>.

[68] RAHUL, R.: *Best Python libraries for Machine Learning*. [online]. [cit. 8. 12. 2022]. Dostupné z URL: <https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/>.

[69] BOESCH, G.: *The 12 Most Popular Computer Vision Tools in 2022*. [online]. [cit. 8. 12. 2022]. Dostupné z URL: <https://viso.ai/computer-vision/the-most-popular-computer-vision-tools/>.

[70] GOYAL, K.: *Top 7 Python NLP Libraries [And Their Applications in 2023]*. [online]. [cit. 8. 12. 2022]. Dostupné z URL: <https://www.upgrad.com/blog/python-nlp-libraries-and-applications/>.

[71] GUPTA, A.: *A Comprehensive Guide on Deep Learning Optimizers*. [online]. [cit. 8. 12. 2022]. Dostupné z URL: <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>.

[72] BROWNLEE, J.: *Difference Between a Batch and an Epoch in a Neural Network*. [online]. [cit. 8. 12. 2022]. Dostupné z URL: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.

[73] GUPTA, S.: *Every machine learning engineer should know about these common loss functions and when to use them*. [online]. [cit. 9. 12. 2022]. Dostupné z URL: <https://builtin.com/machine-learning/common-loss-functions>.

[74] NARKHEDE, S.: *Understanding Confusion Matrix*. [online]. [cit. 15. 4. 2023]. Dostupné z URL: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.

[75] JORDAN, J.: *Evaluating a machine learning model*. [online]. [cit. 1. 4. 2023]. Dostupné z URL: <https://www.jeremyjordan.me/evaluating-a-machine-learning-model/>.

[76] KORSTANJSE, Joos. *The F1 score*. [online]. [cit. 1. 4. 2023]. Dostupné z URL: <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>.

[77] Ippolito, P. P.: *Hyperparameters Optimization*. [online]. [cit. 15. 4. 2023]. Dostupné z URL: <https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d>.

[78] Towards AI.: *How, When, and Why Should You Normalize Standardize Rescale Your Data?*. [online]. [cit. 1. 3. 2022]. Dostupné z URL: <https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff>.

[79] RAMASUBRAMANIAN, K.; SINGH, A.: *Deep Learning Using Keras and TensorFlow*. [cit. 25. 4. 2023]. Dostupné z URL: <https://doi.org/10.1007/978-1-4842-4215-5_11>.

[80] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. (2011): Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830.

[81] CHEN, T.; GUESTRIN, C.: *XGBoost: A Scalable Tree Boosting System*. [online]. [cit. 25. 4. 2023]. Dostupné z URL: <http://arxiv.org/abs/1603.02754>.

[82] GUOLIN, K.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y.: Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu: *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. [online]. [cit. 25. 4. 2023]. Dostupné z URL: <https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html>.

[83] DOROGUSH, A. V.; ERSHOV, V.; GULIN, A.: *CatBoost: gradient boosting with categorical features support*. [online]. [cit. 25. 4. 2023]. Dostupné z URL: <http://learningsys.org/nips17/assets/papers/paper_11.pdf>.

[84] LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, CH. K.: *Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning*. [online]. [cit. 1. 5. 2023]. Dostupné z URL: <http://jmlr.org/papers/v18/16-365.html>.

[85] KARABIBER, F.: *Binary Classification*. [online]. [cit. 1. 5. 2023]. Dostupné z URL: <https://www.learndatasci.com/glossary/binary-classification/>.

[86] FUCHS, M.: *NN – Artificial Neural Network for binary Classification*. [online]. [cit. 5. 5. 2023]. Dostupné z URL: <https://michael-fuchs-python.netlify.app/2021/02/16/nn-artificial-neural-network-for-binary-classification/#introduction>.

[87] FUCHS, M.: *NN – Artificial Neural Network for binary Classification*. [online]. [cit. 9. 12. 2022]. Dostupné z URL: <https://michael-fuchs-python.netlify.app/2021/02/16/nn-artificial-neural-network-for-binary-classification/>.

[88] FUCHS, M.: *Computer Vision - Convolutional Neural Network*. [online]. [cit. 9. 12. 2022]. Dostupné z URL: <https://michael-fuchs-python.netlify.app/2021/01/08/computer-vision-convolutional-neural-network/>.

[89] Stack Overflow: *How to drop columns which have same values in all rows via pandas or spark dataframe?*. [online]. [cit. 9. 12. 2022]. Dostupné z URL: <https://stackoverflow.com/questions/39658574/how-to-drop-columns-which-have-same-values-in-all-rows-via-pandas-or-spark-dataf>.

[90] TUSHKANOVA, O.; LEVSHUN, D.; BRANITSKIY, A.; FEDORCHENKO, E.; NOVIKOVA, E.; KOTENKO, I.: *Detection of Cyberattacks and Anomalies in Cyber-Physical Systems: Approaches, Data Sources, Evaluation*. [online]. [cit. 15. 5. 2023]. Dostupné z URL: <https://stackoverflow.com/questions/39658574/how-to-drop-columns-which-have-same-values-in-all-rows-via-pandas-or-spark-dataf>.

Seznam symbolů a zkratek

AI	Umělá inteligence – Artificial Intelligence
AKI ČR	Asociace kritické infrastruktury České republiky
CC	Regulátor chlazení – Cooling Controller
CIA	Důvěrnost, integrita, dostupnost – Confidentiality, Integrity, Availability
CISA	Agentura pro kybernetickou bezpečnost a bezpečnost infrastruktury – Cybersecurity and Infrastructure Security Agency
CP	Řídící parametr — Control Parameter
CV	Řídící proměnná -– Control Variable
CVi	Počítačové vidění – Computer Vision
DCS	Distribuovaný řídící systém – Distributed Control Systém
DL	Hluboké učení – Deep Learning
DoS	Útok narušení služby – Denial of Service
ERP	Plánování podnikových zdrojů – Enterprise Resource Planning
FC	Regulátor průtoku — Flow Controller
FN	Falešně negativní — False Negative
FP	Falešně pozitivní — False Positive
GANs	Generative Adversarial Networks
GUI	Grafické uživatelské rozhraní – Graphical User Interface
HAI	HIL-based Augmented ICS
HMI	Rozhraní člověk-stroj – Human Machine Interface
HTM	Model vodní turbíny – Hydro Turbine Model
ICS	Průmyslový řídicí systém – Industrial Control System
IDS	Systém pro odhalení průniku – Intrusion Detection Systém
IIoT	Průmyslový internet věcí – Industrial Internet of Things

LC	Regulátor hladiny — Level Controller
LSTM	Typ neuronové sítě – Long Short-Term Memory
LT	Dlouhý — Long-Term
MES	Výrobní informační systém – Manufacturing Execution Systém
ML	Strojové učení – Machine Learning
NLP	Zpracování přirozeného jazyka – Natural Language Processing
NN	Neuronové sítě – Neural Networks
NLTK	Python knihovna pro účel NLP – Natural Language Toolkit
PC	Regulátor tlaku -– Pressure Controller
PID	Proportional Integral Derivative
PLC	Programovatelný logický automat – Programmable Logic Controller
PV	Regulovaná proměnná — Process Variable
RNN	Rekurentní neuronová síť – Recurrent Neural Networks
SC	Regulátor rychlosti -– Speed Controller
SCADA	Systém pro dohled, řízení a sběr dat – Supervisory Control and Data Acquisition
SP	Referenční hodnota — Setpoint
ST	Krátký -– Short-Term
STM	Model parní turbíny — Steam Turbine Model
SVM	Metoda podpůrných vektorů – Support Vector Machines
TC	Regulátor teploty — Temperature Controller
TN	Pravdivě negativní – True Negative
TP	Pravdivě pozitivní – True Positive

Seznam příloh

A Detailní popis prvků datové sady HAI 22.04

Tab. A.1: Datové body v rámci datové sady HAI 22.04 (první část) [58]

No.	Název	Rozsah		Jednotka	Popis
		Min.	Max.		
1	P1_B2004	0	10	bar	Referenční hodnota tlaku výměníku tepla
2	P1_B2016	0	10	bar	SP tlaku pro regulaci tepelného výkonu
3	P1_B3004	0	720	mm	Hladina vody (nádrž na vratnou vodu)
4	P1_B3005	0	2500	l/h	SP průtoku na výstupu (nádrž na vratnou vodu)
5	P1_B4002	0	100	°C	SP výstupní teploty výměníku tepla
6	P1_B4005	0	100	%	Výstup regulace teploty PID
7	P1_B400B	0	2500	l/h	SP odtoku vody (nádrž s ohřívající se vodou)
8	P1_B4022	0	40	°C	Požadovaná teplota pro regulaci tepelného výkonu
9	P1_FCV01D	0	100	%	Příkaz k nastavení polohy pro ventil FCV01
10	P1_FCV01Z	0	100	%	Aktuální poloha ventilu FCV01
11	P1_FCV02D	0	100	%	Příkaz k nastavení polohy pro ventil FCV02
12	P1_FCV02Z	0	100	%	Aktuální poloha ventilu FCV02
13	P1_FCV03D	0	100	%	Příkaz k nastavení polohy pro ventil FCV03
14	P1_FCV03Z	0	100	%	Aktuální poloha ventilu FCV03
15	P1_FT01	0	2500	mmH2O	Naměřený průtok v nádrži na vratnou vodu
16	P1_FT01Z	0	3190	l/h	Přítok vody převedené z P1_FT01
17	P1_FT02	0	2500	mmH2O	Naměřený průtok v nádrži s ohřívající se vodou
18	P1_FT02Z	0	3190	l/h	Odtok vody převedené z P1_FT02
19	P1_FT03	0	2500	mmH2O	Naměřený průtok v nádrži na vratnou vodu
20	P1_FT03Z	0	3190	l/h	Rychlost odtoku vody převedené z P1_FT03
21	P1_LCV01D	0	100	%	Příkaz k nastavení polohy pro ventil LCV01
22	P1_LCV01Z	0	100	%	Aktuální poloha ventilu LCV01
23	P1_LIT01	0	720	mm	Hladina vody v nádrži na vratnou vodu
24	P1_PCV01D	0	100	%	Příkaz k nastavení polohy ventilu PCV01
25	P1_PCV01Z	0	100	%	Aktuální poloha ventilu PCV01
26	P1_PCV02D	0	100	%	Příkaz k nastavení polohy ventilu PCV2
27	P1_PCV02Z	0	100	%	Aktuální poloha ventilu PCV02
28	P1_PIT01	0	10	bar	Výstupní tlak výměníku tepla
29	P1_PIT01_HH	0	10	bar	Nejvyšší výstupní tlak výměníku tepla
30	P1_PIT02	0	10	bar	Tlak vody v čerpadle ohřáté vody
31	P1_PP01AD	0	1	Boolean	Příkaz ke spuštění hlavního vodního čerpadla PP01A
32	P1_PP01AR	0	1	Boolean	Provozní stav hlavního vodního čerpadla PP01A
33	P1_PP01BD	0	1	Boolean	Příkaz ke spuštění hlavního vodního čerpadla PP01B
34	P1_PP01BR	0	1	Boolean	Provozní stav hlavního vodního čerpadla PP01B
35	P1_PP02D	0	1	Boolean	Příkaz ke spuštění čerpadla ohřáté vody PP02
36	P1_PP02R	0	1	Boolean	Provozní stav čerpadla ohřáté vody PP02
37	P1_PP04	0	100	%	Ovládání čerpadla chladiče
38	P1_PP04SP	0	100	°C	SP teploty chladiče
39	P1_SOL01D	0	1	Boolean	Příkaz k otevření přívodního ventilu nádrže na vodu
40	P1_SOL03D	0	1	Boolean	Příkaz k otevření vypouštěcího ventilu nádrže na vodu
41	P1_STSP	0	1	Boolean	Příkaz ke spuštění/zastavení bojleru DCS
42	P1_TIT01	-50	150	°C	Výstupní teplota výměníku tepla
43	P1_TIT02	-50	150	°C	Teplota nádrže s ohřívající se vodou

Tab. A.2: Datové body v rámci datové sady HAI 22.04 (druhá část) [58]

No.	Název	Rozsah		Jednotka	Popis
		Min.	Max.		
44	P1_TIT03	-50	150	°C	Teplota hlavní nádrže na vodu
45	P2_24Vdc	0	30	Voltage	Vstupní napětí DCS 24V
46	P2_ATSW_Lamp	0	1	Boolean	Lampa automatického SW
47	P2_AutoGo	0	1	Boolean	Tlačítko automatického spuštění
48	P2_AutoSD	0	3200	RPM	Automatický požadavek rychlosti
49	P2_Emerg	0	1	Boolean	Nouzové tlačítko
50	P2_MASW	0	1	Boolean	Manuální(1)/automatický(0) SW
51	P2_MASW_Lamp	0	1	Boolean	Lampa manuálního SW
52	P2_ManualGO	0	1	Boolean	Tlačítko ručního spuštění
53	P2_ManualSD	0	3200	RPM	Manuální požadavek rychlosti
54	P2_OnOff	0	1	Boolean	Spínač zapnutí/vypnutí turbíny DCS
55	P2_RTR	0	2880	RPM	Otáčky za minutu
56	P2_SCO	0	100000	–	Ovládání výstupní hodnoty regulátoru otáček
57	P2_SCST	-100	100	RPM	Změna rychlosti úměrná změně frekvence STM
58	P2_SIT01	0	3200	RPM	Aktuální otáčky turbíny měřené otáčkovou sondou
59	P2_TripEx	0	1	Boolean	Tlačítko nouzového východu
60	P2_VIBTR01	-10	10	µm	Posunutí v ose Y související s vibracemi 1. hřídele
61	P2_VIBTR02	-10	10	µm	Posunutí v ose X související s vibracemi 1. hřídele
62	P2_VIBTR03	-10	10	µm	Posunutí v ose Y související s vibracemi 2. hřídele
63	P2_VIBTR04	-10	10	µm	Posunutí v ose X související s vibracemi 2. hřídele
64	P2_VT01	11	12	rad/s	Signál fázového zpoždění klíčové fázorové sondy
65	P2_VTR01	-10	10	µm	Přednastavený limit vibrací pro snímač P2_VIBTR01
66	P2_VTR02	-10	10	µm	Přednastavený limit vibrací pro snímač P2_VIBTR02
67	P2_VTR03	-10	10	µm	Přednastavený limit vibrací pro snímač P2_VIBTR03
68	P2_VTR04	-10	10	µm	Přednastavený limit vibrací pro snímač P2_VIBTR04
69	P3_FIT01	0	27648	–	Průtok vody přitékající do horní nádrže na vodu
70	P3_LCP01D	0	27648	–	Příkaz k otáčkám čerpadla LCP01
71	P3_LCV01D	0	27648	–	Příkaz k nastavení polohy pro ventil LCV01
72	P3_LH01	0	70	%	SP hladiny vody v horní nádrži
73	P3_LIT01	0	90	%	Hladina vody v horní nádrži na vodu
74	P3_LL01	0	70	%	SP hladiny vody ve spodní nádrži
75	P3_PIT01	0	27648	–	Tlak vody tekoucí do horní nádrže na vodu
76	P4_HT_FD	-0,02	0,02	mHz	Frekvenční odchylka HTM
77	P4_HT_PO	0	100	MW	Výstupní výkon HTM
78	P4_HT_PS	0	100	MW	Plánovaná potřeba energie HTM
79	P4_LD	0	500	MW	Celková poptávka po elektrickém zatížení
80	P4_ST_FD	-0,02	0,02	Hz	Frekvenční odchylka STM
81	P4_ST_GOV	0	27648	–	Rychlost otevírání brány STM
82	P4_ST_LD	0	500	MW	Požadavek na elektrické zatížení STM
83	P4_ST_PO	0	500	MW	Výstupní výkon STM
84	P4_ST_PS	0	500	MW	Plánovaná spotřeba energie STM
85	P4_ST_PT01	0	27648	–	Digitální hodnota tlaku páry STM
86	P4_ST_TT01	0	27648	–	Digitální hodnota teploty páry STM

Tab. A.3: Popis útoků v rámci datové sady HAI 22.04 (první část) [58]

Scénář	Cíl			Popis
	Regulátor	Veličina	Název	
AP01	P1-PC	SP1	P1_B2016	Snížení/zvýšení hodnoty SP P1-PC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP02		SP1	P1_B2016	
		PV1	P1_PIT01	Pokus o udržení předchozí hodnoty senzoru.
AP03		SP1	P1_B2016	Snížení/zvýšení hodnoty SP P1-PC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
		PV1	P1_PIT01	Pokus o udržení předchozí hodnoty senzoru.
		PV2	P1_FIT01	
AP04		CV1	P1_PCV01D	Snížení/zvýšení hodnoty CV P1-PC. Návrat k normě.
AP05				
		PV1	P1_PIT01	Pokus o udržení předchozí hodnoty senzoru.
AP07		CV1-ST	P1_PCV01D	Krátký útok, který na několik sekund sníží/zvýší hodnotu CV P1-PC a poté vrátí normální hodnotu. Opakuje se několikrát a změny SP v HMI jsou skryty.
AP08	P1-FC	SP1	P1_B3005	Snížení/zvýšení hodnoty SP P1-FC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP09				
		PV1	P1_FT03	Pokus o udržení předchozí hodnoty senzoru.
AP10		SP1	P1_B3005	Snížení/zvýšení hodnoty SP P1-FC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
		PV1	P1_FT03	Pokus o udržení předchozí hodnoty senzoru.
		PV2	P1_LIT01	
AP11		CV1	P1_FCV03D	Snížení/zvýšení hodnoty SP P1-FC. Obnovení v podobě lichoběžníkového profilu.
AP12		CV1	P1_FCV03D	Snížení/zvýšení hodnoty CV P1-FC. Návrat k normě.
		PV1	P1_FT03	Pokus o udržení předchozí hodnoty senzoru.
AP13	P1-LC	CV1-ST	P1_FCV03D	Krátký útok, který na několik sekund sníží/zvýší hodnotu CV P1-FC a poté vrátí normální hodnotu. Opakuje se několikrát a změny SP v HMI jsou skryty.
AP14		SP1	P1_B3004	Snížení/zvýšení hodnoty SP P1-LC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP15				
		PV1	P1_LIT01	Pokus o udržení předchozí hodnoty senzoru.
AP16		CV1	P1_LCV01D	Snížení/zvýšení hodnoty CV P1-LC. Návrat k normě.
AP17				
		PV1	P1_LIT01	Pokus o udržení předchozí hodnoty senzoru.
AP18		CV1-ST	P1_LCV01D	Krátký útok, který na několik sekund sníží/zvýší hodnotu CV P1-LC a poté vrátí normální hodnotu. Opakuje se několikrát a změny SP v HMI jsou skryty.

Tab. A.4: Popis útoků v rámci datové sady HAI 22.04 (druhá část) [58]

Scénář	Cíl			Popis
	Regulátor	Veličina	Název	
AP19	P1-TC	CV1	P1_FCV01D	Snížení/zvýšení hodnoty CV P1-TC. Návrat k normě.
AP20		PV1	P1_TIT01	Pokus o udržení předchozí hodnoty senzoru.
AP21		CV1-ST	P1_FCV01D	Krátký útok, který na několik sekund sníží/zvýší hodnotu CV P1-TC a poté vrátí normální hodnotu. Opakuje se několikrát a změny SP v HMI jsou skryty.
AP22		SP1-LT	P1_B4002	Dlouhý útok, který snižuje nebo zvyšuje hodnotu SP P1-TC nepřetržitě po dobu delší než 10 minut a poté vrátí normální hodnotu.
AP23	P1-CC	CC1	P1_PP04	Snížení/zvýšení hodnoty CV P1-CC. Návrat k normě.
AP24		CV1-ST	P1_PP04	Krátký útok, který na několik sekund sníží/zvýší hodnotu CV P1-CC a poté vrátí normální hodnotu. Opakuje se několikrát a změny SP v HMI jsou skryty.
AP25		SP1-LT	P1_PP04_SP	Dlouhý útok, který snižuje nebo zvyšuje hodnotu SP P1-CC nepřetržitě po dobu delší než 10 minut a poté vrátí normální hodnotu.
AP26	P2-SC	SP1	P2_AutoSD	Snížení/zvýšení hodnoty SP P2-SC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP27		PV1	P2_SIT01	Pokus o udržení předchozí hodnoty senzoru.
AP28		SP2	P2_ManualSD	Snížení/zvýšení hodnoty SP P2-FC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP29		CV1	P2_SCO	Snížení/zvýšení hodnoty CV P2-SC. Návrat k normě.
AP30		PV1	P2_SIT01	Pokus o udržení předchozí hodnoty senzoru.
AP31		SP1-ST	P2_AutoSD	Krátký útok, který na několik sekund sníží/zvýší hodnotu CV P2-SC a poté vrátí normální hodnotu. Opakuje se několikrát a změny SP v HMI jsou skryty.
AP33	P2-TC	SP2	P2_VTR02	Snížení/zvýšení hodnoty SP P2-SC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP34		SP3	P2_RTR	Snížení/zvýšení hodnoty SP P2-SC. Obnovení v podobě lichoběžníkového profilu při skrytí změn SP v HMI.
AP35	P3-LC	CV1	P3_LCP01D	Pokus o udržení předchozí hodnoty senzoru.
AP36				Snížení/zvýšení hodnoty CV P3-LC. Návrat k normě.
AP37		PV1	P3_LIT01	Pokus o udržení předchozí hodnoty senzoru.
AP38		CV2	P3_LCV01D	Snížení/zvýšení hodnoty CV P3-LC. Návrat k normě.
AP39		PV1	P3_LIT01	Pokus o udržení předchozí hodnoty senzoru.
		CV2-LT	P3_LCV01D	Dlouhý útok, který snižuje nebo zvyšuje hodnotu SP P3-LC nepřetržitě po dobu delší než 10 minut a poté vrátí normální hodnotu.

Tab. A.5: Popis trvání útoků v rámci datové sady HAI 22.04 (první část) [58]

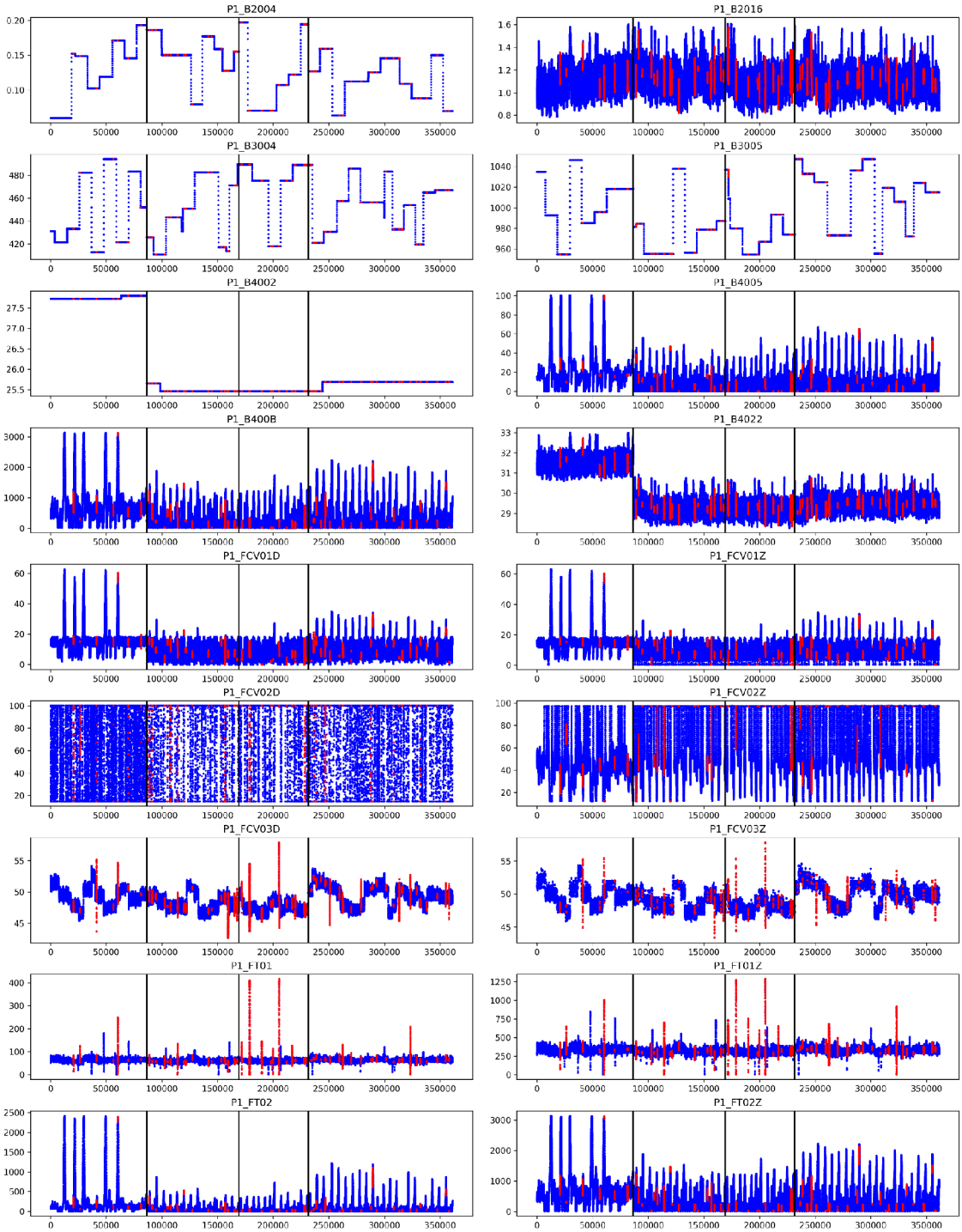
No.	ID	Útoky			Počáteční čas		Trvání [s]
		Scénář	Cílový regulátor	Cíl(e) útoku			
1	A101	AP04	P1-PC-CO1	P1_PCV01D	10.07.2021	5:41	190
2	A102	AP18	P1-LC-CO1-ST	P1_LCV01D		7:19	54
3	A103	AP11	P1-FC-CO1PV1	P1_FCV03D, P1_FT03		11:25	126
4	A104	AP37	P3-LC-CO2	P3_LCV01D		15:39	54
5	A105	AP14	P1-LC-SP1	P1_B3004		16:42	296
6	A106	AP13	P1-CC-CO1	P1_PP04		19:21	91
7	A107	AP19	P1-TC-CO1	P1_FCV01D		22:35	67
8	A201	AP01	P1-PC-SP1	P1_B2016	13.07.2021	16:38	257
9	A202	AP13	P1-FC-CO1-ST	P1_FCV03D		17:21	65
10	A203	AP31	P2-SC-SP1-ST	P2_AutoSD		18:13	45
11	A204	AP04	P1-PC-CO1	P1_PCV01D		20:28	248
		AP29	P2-SC-CO1	P2_SCO			
12	A205	AP37	P3-LC-CO2	P3_LCV01D		21:10	55
13	A206	AP02	P1-PC-SP1PV1	P1_B2016, P1_PIT01		21:58	176
		AP27	P2-SC-SP1PV1	P2_AutoSD, P2_SIT01			
14	A207	AP16	P1-LC-CO1	P1_LCV01D		23:40	284
15	A208	AP30	P2-SC-CO1PV1	P2_SCO, P2_SIT01	14.07.2021	1:15	152
16	A209	AP03	1-PC SP1PV1PV2	P1_B2016, P1_PIT01, P1_FIT01		1:40	162
17	A210	AP26	P2-SC-SP1	P2_AutoSD		3:23	97
18	A211	AP05	P1-PC- CO1PV1	P1_PCV01D, P1_PIT01		7:21	151
19	A212	AP35	P3-LC-CO1	P3_LCP01D		8:11	55
20	A213	AP24	P1-CC-CO1-ST	P1_PP04		10:35	80
21	A214	AP39	P3-LC-CO2-LT	P3_LCV01D		11:23	613
22	A215	AP09	P1-FC-SP1PV1	P1_B3005, P1_FT03		12:17	168
23	A216	AP01	P1-PC-SP1	P1_B2016		13:52	158
		AP08	P1-FC-SP1	P1_B3005			
24	A217	AP10	P1-FC-CO1	P1_FCV03D		14:31	98
25	A301	AP16	P3-LC-CO2	P2_LCV01D		18:21	348
		AP10	P1-FC-CO1	P1_FCV03D			
26	A302	AP15	P1-LC-SP1PV1	P1_LCV01D		20:16	358
27	A303	AP17	P1-LC-CO1PV1	P1_B3004. P1_LIT01		23:22	143
		AP37	P3-LC-CO2	P3_LCV01D			
28	A304	AP38	P3-LC-CO2PV1	P1_LCV01D. P1_LIT01	15.07.2021	1:41	91
29	A305	AP18	P1-LC-CO1-ST	P3_LCV01D		2:09	94
30	A306	AP04	P1-PC-CO1	P3_LCV01D		3:37	353
		AP15	P1-LC-SP1PV1	P1_B3004. P1_LIT01			
31	A307	AP20	P1-TC-CO1PV1	P1_FCV01D. P1_TIT01		5:35	151
32	A308	AP05	P1-PC-CO1PV1	P1_PCV01D, P1_PIT01		6:53	173
		AP23	P1-CC-CO1	P1_PP04			
33	A309	AP08	P1-FC-SP1	P1_B3005		7:42	96
		AP19	P1-TC-CO1	P1_FCV01D			

Tab. A.6: Popis trvání útoků v rámci datové sady HAI 22.04 (druhá část) [58]

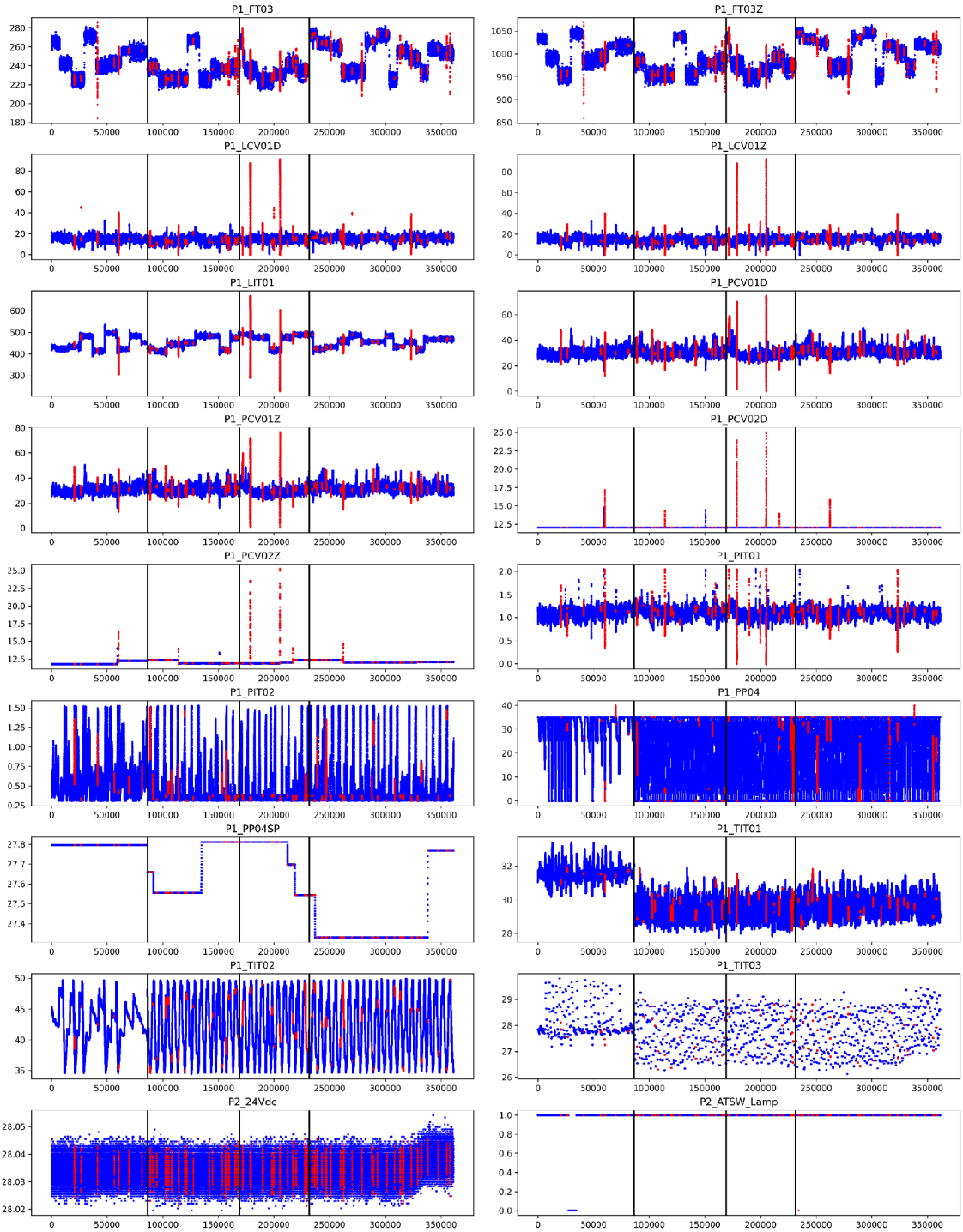
No.	ID	Útoky			Počáteční čas		Trvání [s]
		Scénář	Cílový regulátor	Cíl(e) útoku			
34	A310	AP35	P3-LC-CO1	P3_LCP01D	15.07.2021	9:52	2024
		AP37	P3-LC-CO2	P3_LCV01D			
35	A401	AP28	P2-SC-SP2	P2_ManualSD		12:42	38
36	A402	AP21	P1-TC-CO1-ST	P1_FCV01D		13:20	88
37	A403	AP34	P2-TC-SP3	P2_RTR		13:57	96
38	A404	AP26	P2-SC-SP1	P2_AutoSD		15:08	97
		AP37	P3-LC-CO2	P3_LCV01D			
39	A405	AP22	P1-TC-SP1-LT	P1_B4002		16:07	505
40	A406	AP09	P1-FC-SP1PV1	P1_B3005, P1_FT03		17:22	186
		AP19	P1-TC-CO1	P1_FCV01D			
41	A407	AP13	P1-FC-CO1-ST	P1_FCV03D		19:45	122
		AP17	P1-LC-CO1PV1	P1_LCV01D, P1_LIT01			
42	A408	AP05	P1-PC-CO1PV1	P1_PCV01D, P1_PIT01		20:29	673
		AP17	P1-LC-CO1PV1	P1_LCV01D, P1_LIT01			
43	A409	AP18	P1-LC-CO1-ST8	P1_LCV01D		22:41	63
		AP21	P1-TC-CO1-ST9	P1_FCV01D			
44	A410	AP11	P1-FC-CO1PV1	P1_FCV03D, P1_FT03	16.07.2021	1:07	179
		AP27	P2-SC-SP1PV1	P2_AutoSD, P2_SIT01			
45	A411	AP23	P1-CC-CO1	P1_PP04		3:35	99
		AP34	P2-TC-SP3	P2_RTR			
46	A412	AP20	P1-TC-CO1PV1	P1_FCV01D, P1_TIT01		4:02	156
		AP01	P1-PC-SP1	P1_B2016			
47	A413	AP16	P1-LC-CO1	P1_LCV01D		4:59	153
		AP27	P2-SC-SP1PV1	P2_AutoSD, P2_SIT01			
48	A414	AP33	P2-TC-SP2	P2_VTR02		7:20	77
		AP36	P3-LC-CO1PV1	P3_LCP01D, P3_LIT01			
49	A415	AP3	P2-TC-SP2	P2_VTR02		9:17	77
50	A416	AP12	1-FC CO1PV1PV2	P1_FCV03D, P1_FT03, P1_LIT01		10:39	134
51	A417	AP25	P1-CC-SP1-LT	P1_PP04_SP.		11:22	544
52	A418	AP01	P1-PC-SP1	P1_B2016		13:23	342
		AP14	P1-LC-SP1	P1_B3004			
53	A419	AP01	P1-PC-SP1	P1_B2016		14:59	163
		AP35	P3-LC-CO1	P3_LCP01D			
54	A420	AP07	P1-PC-CO1-ST	P1_PCV01D		15:57	89
55	A421	AP30	P2-SC-CO1PV1	P2_SCO, P2_SIT01		17:34	152
		AP23	P1-CC-CO1	P1_PP04			
56	A422	AP02	P1-PC-SP1PV1	P1_B2016, P1_PIT01		20:08	165
		AP26	P2-SC-SP1	P2_AutoSD			
57	A423	AP08	P1-FC-SP1	P1_B3005		22:17	115
		AP29	P2-SC-CO1	P2_SCO			
58	A424	AP10	P1-FC-CO1	P1_FCV03D		23:05	86
		AP23	P1-CC-CO1	P1_PP04			

B Vykreslení datových bodů

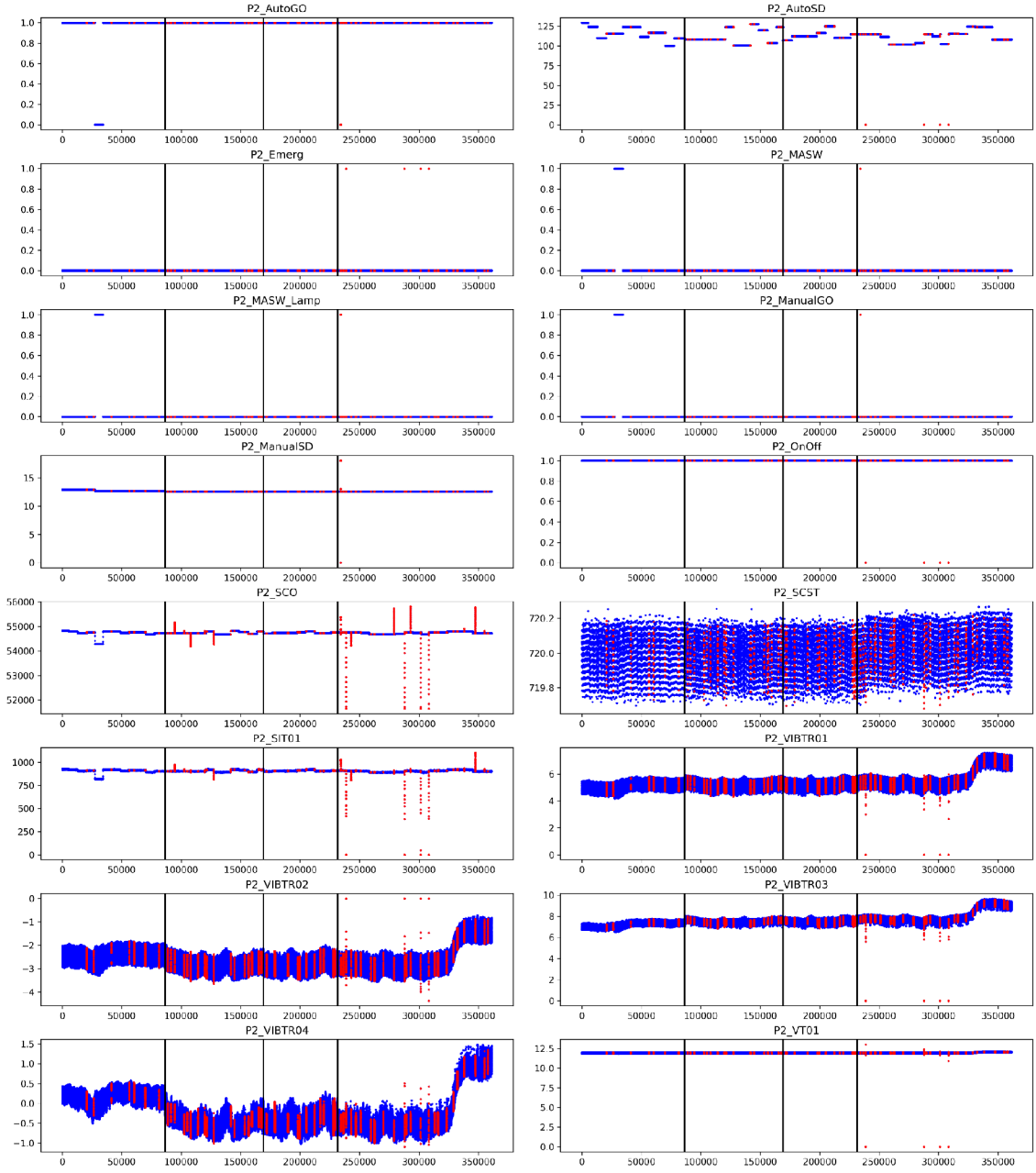
V této části příloh jsou vykresleny všechny hodnoty datových bodů souborů *test1.csv*, *test2.csv*, *test3.csv* a *test4.csv* (v grafech odděleny černými, vertikálními linkami).



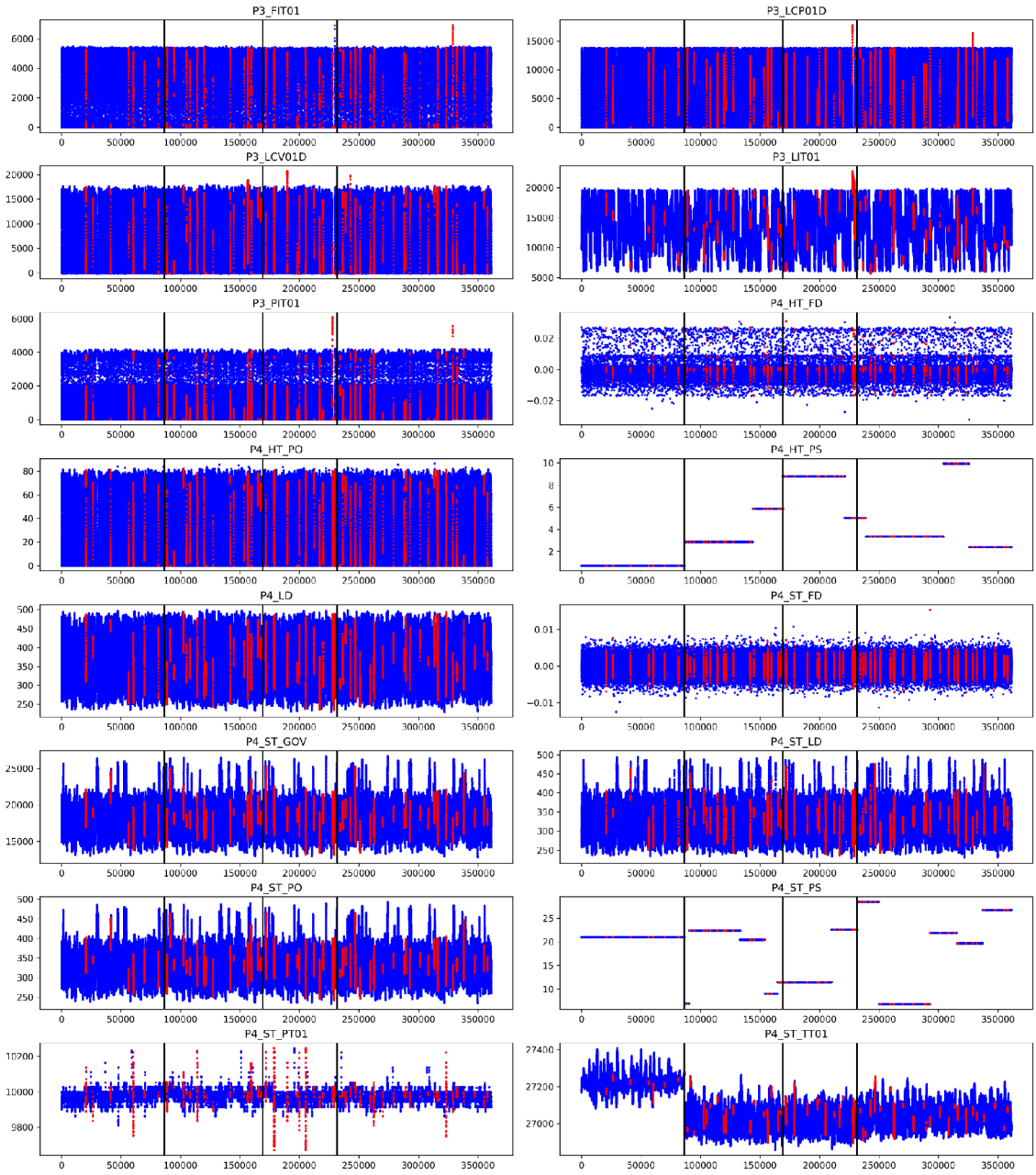
Obr. B.1: Vykreslení datových bodů datové sady HAI 22.04 (první část)



Obr. B.2: Vykreslení datových bodů datové sady HAI 22.04 (druhá část)



Obr. B.3: Vykreslení datových bodů datové sady HAI 22.04 (třetí část)



Obr. B.4: Vykreslení datových bodů datové sady HAI 22.04 (čtvrtá část)

C Obsah elektronické přílohy

- `/` ..... kořenový adresář přiloženého archivu
  - `colab` ..... Vyhotovené zdrojové kódy v sešitech Google Colab
    - `ML_supervised` ..... Strojové učení s učitelem
      - `HAI_ML_1.ipynb` ..... Výchozí řešení
      - `HAI_ML_2.ipynb` ..... Vyvážení trénovací datové sady
      - `HAI_ML_3.ipynb` ..... Změna poměrů datových sad
      - `HAI_ML_4.ipynb` ..... Ladění hyperparametrů
      - `HAI_ML_5.ipynb` ..... Odstranění datových bodů
      - `HAI_ML_6.ipynb` ..... Promíchání dat
    - `NN` ..... Neuronové sítě
      - `HAI_NN_1.ipynb` ..... Výchozí řešení
      - `HAI_NN_2.ipynb` ..... Vyvážení trénovací datové sady
      - `HAI_NN_3.ipynb` ..... Změna poměrů datových sad
      - `HAI_NN_4.ipynb` ..... Odstranění datových bodů
      - `HAI_NN_5.ipynb` ..... Feature importance
      - `HAI_NN_6.ipynb` ..... Promíchání dat
    - `ML_unsupervised` ..... Strojové učení bez učitele
      - `HAI_ML_bezUcitele.ipynb` ..... One-class SVM