Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Disertační práce

Návrh a ověření systému detekce anomálií založeného na strojovém učení v průmyslových řídicích systémech

Design and verification of anomaly detection system based on machine learning in industrial control systems

Autor: **Ing. Jan Vávra**

Studijní program: Inženýrská informatika P3902
Studijní obor: Inženýrská informatika 3902V023

Školitel: doc. Ing. Luděk Lukáš, CSc.
Konzultant: doc. Ing. Martin Hromada, Ph.D.

Zlín, prosinec 2020

Klíčová slova: *kybernetická bezpečnost, strojové učení, umělá inteligence, detekce anomálií, průmyslový řídicí systém.*

Key words: *Cyber Security, Machine Learning, Artificial Intelligence, Anomaly Detection, Industrial Control System.*

PODĚKOVÁNÍ

Rád bych poděkoval svému školiteli doc. Ing. Luďkovi Lukášovi, CSc. a konzultantovi doc. Ing. Martinovi Hromadovi, Ph.D. za odborné vedení v průběhu celého studia, cenné rady a připomínky při řešení disertační práce. Rád bych poděkoval i své rodině za morální podporu v průběhu mého studia.

Dále bych chtěl poděkovat A.I.Lab na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně (ailab.fai.utb.cz) za propůjčené prostředky. Také bych chtěl poděkovat mezinárodním pracovištím za poskytnutí datasetů, bez nichž by tato práce nemohla vzniknout. A v neposlední řadě je velice oceňován přístup k výpočetním a skladovacím zařízením vlastněným stranami a projekty přispívajícími do národní sítě gridové infrastruktury MetaCentrum poskytované v rámci programu „Projekty velkých infrastruktur výzkumu, vývoje a inovací“ (CESNET LM2015042), které přispělo k dokončení této disertační práce.

ABSTRAKT

Technologie se staly nedílným prvkem současné společnosti. Současný přechod od průmyslové společnosti k informační společnosti je doprovázen implementací nových technologií do každé části lidské činnosti. Zvyšující se tlak na aplikaci informačních a komunikačních technologií v oblastech kritické infrastruktury a jejich řídicích systémů, zapříčiňuje vznik nových zranitelností. Tradiční přístupy pro zajištění bezpečnosti se stávají neefektivními. Z tohoto pohledu je využití umělé inteligence další evolučním krokem, který poskytuje robustní řešení i pro velmi rozsáhlé a komplexní systémy. Tato disertační práce je zaměřena na oblast výzkumu v rámci kybernetické bezpečnosti pro průmyslové řídicí systémy, které jsou četně využívány v kritické infrastruktuře. Kybernetická bezpečnost průmyslových řídicích systémů je z tohoto pohledu jedním z velmi důležitých druhů bezpečnosti pro fungování moderního státu. Hlavním jádrem disertační práce je vytvoření systému detekce anomálií, založeného na metodách strojového učení ve specifické oblasti kybernetické bezpečnosti průmyslových řídicích systémů. Zvláštní pozornost je poté věnována optimalizaci zvoleného řešení. Výsledný systém detekce anomálií je vytvořen s ohledem na jeho autonomní provoz a určitou míru interpretace kybernetických útoků.

ABSTRACT:

Technology has become an integral part of contemporary society. The current transition from an industrial society to the information society is accompanied by the implementation of new technologies in every part of human activity. Increasing pressure to apply ICT in critical infrastructure and their control systems creates new vulnerabilities. Traditional safety approaches are becoming ineffective. From this perspective, the use of artificial intelligence is another evolutionary step that provides robust solutions for extensive and sophisticated systems. This dissertation focuses on the field of cybersecurity research for industrial control systems that are widely used in critical information infrastructure. Cybernetic protection for industrial control systems is one of the most important types of security for a modern state. The main core of the thesis is to create an anomaly detection system based on machine learning methods in a specific area of cyber security of industrial control systems. Special attention is then paid to optimization. The resulting anomaly detection system is created concerning its autonomous operation and some degree of cyber-attack interpretation.

OBSAH

1. ÚVOD

Technologie v posledních několika dekádách zaznamenaly exponenciální růst v oblastech nasazení, efektivity a funkcionality. Systémy založené na zmíněných technologiích využívají automatizace, digitalizace, robotizace a jsou často vzájemně propojeny s využitím vzdáleného řízení. Mluví se o revoluci v podobě tzv. „Průmyslu 4.0“, který v budoucnu ovlivní většinu aspektů lidské společnosti. Frank [1] ve své publikaci prezentoval základní atributy, ve kterých je Průmysl 4.0 uplatňován. Jedná se o integraci, řízení energií, vystopovatelnost (tzv. „tracebility“), automatizaci, virtualizaci a flexibilitu pomocí technologií jako je Internet věcí (IoT), „cloud computing“. „big data“ a analytika. Právě dvě posledně jmenované oblasti („big data“ a analytika) jsou podle řady autorů považovány za největší motory průmyslové revoluce 4.0. [1] Do této skupiny řadíme techniky dolování dat („data mining“) a strojového učení („machine learning“).

Mezi jeden ze základních prvků Průmyslu 4.0 patří automatizované a autonomní systémy. Ty jsou využívány jako náhrada klasické lidské činnosti. To má za následek, že jsou komunikačně propojovány i technologické celky, které spadají do oblasti kritické infrastruktury (KI). Z tohoto důvodu jsou systémy KI zatíženy značným tlakem z pohledu relativně nových hrozeb, jakými jsou například kybernetické útoky, které se z historického hlediska stávají zásadní hrozbou pro dnešní a budoucí společnost. Nebezpečnost jakékoliv hrozby je tím markantnější, čím větší jsou dopady při jejím uskutečnění. V tomto ohledu lze považovat KI za potencionálně nejvíce ohroženou oblast, jak z pohledu privátních, tak státních aktérů. Destabilizace a ztráta funkčnosti KI může zapříčinit vážné ohrožení životního prostředí, obyvatelstva, finančního sektoru, popřípadě základních funkcí státu. Zvláště ohrožené jsou řídicí průmyslové systémy (Industrial Control System – ICS), jelikož představují skupinu systémů přímo ovlivňující fyzický svět. Z tohoto důvodu představují velmi ceněný cíl pro mnoho útočníků. Výpadek těchto systémů má za následek ohrožení života a zdraví obyvatelstva, základních služeb společnosti nebo životního prostředí. Zásadní přehodnocení ICS kybernetické bezpečnosti bylo zapříčiněno působením počítačového červa Stuxnet v roce 2010. [2]

Ochrana komplexních systémů, jako je ICS, skýtá nové výzvy pro odborníky v oblasti kybernetické bezpečnosti. Ochránit systémy, které v reálném čase generují statisíce multidimenzionálních záznamů za poměrně krátkou dobu je velice náročný úkol na čas i prostředky. Z tohoto důvodu je vhodné využít technik umělé inteligence v podobě dolování dat a strojového učení.

Vzrůstající hrozba kybernetických útoků je reflektovaná celou řadou států. V rámci Evropské unie (EU), ale i celosvětově, lze vidět zřetelný trend v koncepčním začlenění řešené problematiky kybernetické bezpečnosti mezi strategické cíle státu. Mezi první země EU, které přijaly ucelenou strategii pro

kybernetickou bezpečnost, patří Slovensko, Německo, Francie, Litva, Anglie a také Česká republika (ČR). Přičemž první nasazení CERT (Computer Emergency Response Team) nastalo v Litvě a Lotyšsku. V případě ČR byl CERT zřízen v roce 2011, čímž byla zvýšena schopnost řešení kybernetických incidentů na úrovni státu. Kybernetická bezpečnost ČR je řešena uceleným právním předpisem ve formě zákona č. 181/2014 Sb., o kybernetické bezpečnosti a prováděcích vyhlášek č. 316/2014 Sb., vyhláška o kybernetické bezpečnosti a č. 317/2014 Sb., vyhláška o významných informačních systémech a jejich určujících kritériích, které řeší komplexně otázku kybernetické bezpečnosti a stanovuje dva pilíře zajišťující budoucí kybernetickou bezpečnost v ČR.

Předložená disertační práce je zaměřena na detekci anomálií v oblasti průmyslových řídicích systémů z pohledu kybernetické bezpečnosti. Důraz je kladen na respektování specifik ICS systémů. Ty jsou často rozdílné od požadavků běžně využívané výpočetní techniky. Pro detekci anomálií jsou využity metody, techniky a algoritmy vztahující se ke strojovému učení a dolování dat. Každý ze zvolených postupů je podrobně analyzován v následujících kapitolách. V závislosti na vybraném nastavení a vytvořené hypotéze jsou vyhotoveny experimenty, které hodnotí zvolená řešení.

2. ZHODNOCENÍ SOUČASNÉHO STAVU

Kybernetická bezpečnost se stala v uplynulých deseti letech jednou z hlavních bezpečnostních otázek a problémů nejenom v tzv. „západní civilizaci“, do které patří například Evropa nebo Spojené státy americké (USA), ale i pro zbytek světa. Tento trend je popsán značnou řadou autorů [3], [4], [5], [6].

„*Jedním z nejvíce problematických elementů kybernetické bezpečnosti je rychlý a neustále se vyvíjející charakter útoků, rizik a hrozeb. Bezpečnostní otázky musí být řízeny za účelem zajištění přežití a prosperování organizace*.“ [6]

Z tohoto důvodu je nutné kybernetické hrozby vnímat nejenom na národní, ale především na globální úrovni. Proto je zde potřeba národních i nadnárodních organizací zabývajících se kybernetickou bezpečností. Mezi takové organizace lze zařadit organizaci European Union Agency for Network and Information Security (ENISA), která je pomyslným centrem pro sdílení odborných znalostí, týkajících se kybernetické bezpečnosti v rámci EU.

Opomenout nelze značný počet organizací a asociací, zabývajících se kybernetickou bezpečností. Mezi nejvýznamnější patří: Information Systems Audit and Control Association (ISACA), Information Technology Infrastructure Library (ITIL), The SANS Institute, International Information Systems Security Certification Consortium (ISC), Forum of Incident Response and Security Teams (FIRST), Information Systems Security Association (ISSA), Center for Internet Security (CIS), National Association of ISACs, National Cyber Security Alliance.

Oblast kybernetické bezpečnosti je v současnosti dynamický fenomén, který zásadně ovlivňuje funkce současné společnosti. Z dosavadního vývoje lze predikovat nárůst kybernetických incidentů v následujících letech z důvodu digitalizace a propojování funkcí společností. Již v dnešní době mluvíme o tzv. „Průmyslu 4.0“, který označuje po první průmyslové revoluci – industrializace, druhé průmyslové revoluci – elektrifikace, třetí průmyslové revoluci – automatizace, již čtvrtou průmyslovou revoluci, která je založena na digitalizaci a masivním rozšíření Internetu. Tato revoluce je ze značné části založena na Internetu věcí, Internetu služeb a Digitalizaci ekonomiky, což povede ke zvýšení efektivity fungování lidské společnosti. Tento přerod nebude poskytovat jenom benefity, ale také zapříčiní dosud nevídaný rozvoj kybernetických incidentů v důsledku rozšiřování kybernetického prostoru do každého aspektu lidské činnosti.

2.1 PRŮMYSLOVÉ ŘÍDICÍ SYSTÉMY

Průmyslové řídicí systémy (ICS) jsou systémy navrhnuté pro podporu, řízení a kontrolu průmyslových procesů. Ty jsou často součástí kritické infrastruktury, kde pronikají do oblastí dopravních systémů, elektráren, přehrad, zpracování

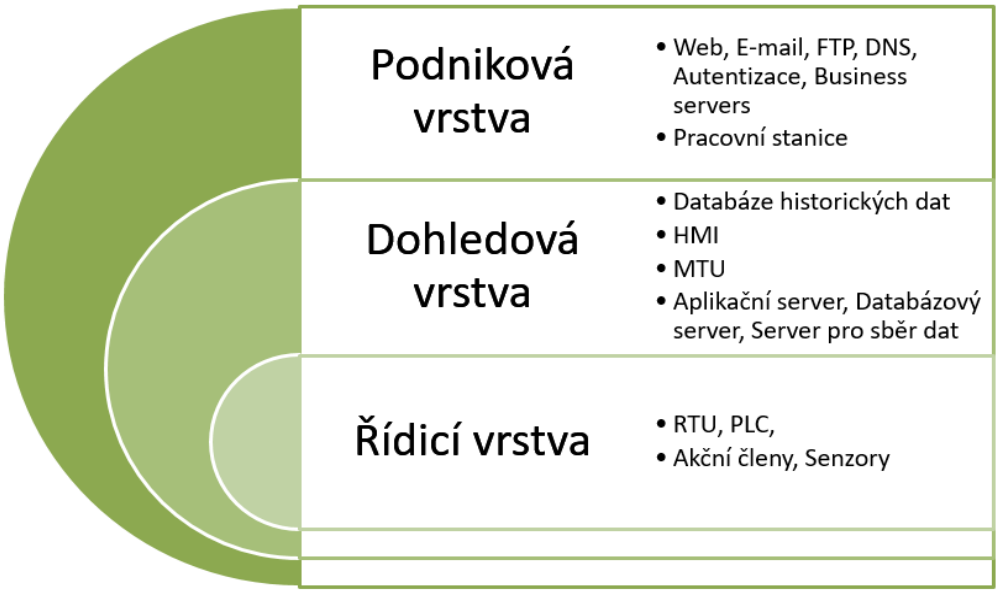
vody, výroby oleje, chemikálií, distribuce plynu atd. Podle publikace „Guide to Industrial Control Systems (ICS) Security“ [7], vydané institucí NIST (National Institute of Standards and Technology), lze rozdělit ICS do dvou základních podskupin. První z nich je geograficky nezávislý „Supervisory Control and Data Acquisition“ (SCADA) systém. Do druhé skupiny je zařazen geograficky závislý systém „Distributed Control System“ (DCS).[8] Hranice mezi těmito systémy bývá často poměrně špatně definovaná, což vede ke vzájemné záměně jednotlivých skupin. Navíc lze konstatovat, že značná část odborné veřejnosti využívá akronym SCADA i pro oblast DCS. Souhrn hlavních rozdílů mezi systémy SCADA a DSC je uveden v Tab. 1 - Porovnání SCADA a DCS systémů. [8], [9]. Toto rozdělení je reflektováno také autorem Ginter [9], který definuje SCADA a DCS systémy jako technologii se stejnou funkcionalitou, avšak rozdílnou geografickou vzdáleností od průmyslových procesů a zařízení. Z tohoto důvodu je vhodné využívat zavedené terminologie akceptované velkou částí vědecké komunity, tedy ICS, popřípadě SCADA.

Tab. 1 - Porovnání SCADA a DCS systémů. [8], [9]

Porovnané systémy/Charakteristika systému	***SCADA***	***DCS***
Zaměření	Sběr dat	Trendy procesů
Geografické rozložení	Globální	Lokální
Získávání nezbytných dat	Využití databáze	Využití I/O zařízením
Účel	Koordinace infrastruktury	Kontrola koncových zařízení
Rozmístění	Centralizované	Decentralizované

Pomyslná specifikace hierarchických skupin systému vykazuje v případě ICS ale i ICT poměrně variabilní povahu. Je nutno poznamenat, že jednotlivá řešení ICS a ICT systémů jsou často poměrně rozdílná a liší se v závislosti na řešeném případu. Avšak z pohledu kybernetické bezpečnosti jsou definovány tři základní oblasti: podniková vrstva, dohledová vrstva a řídicí vrstva. Dohledová a řídicí vrstva je klasickou složkou každého ICS. V současné době je však účelné uvažovat o podnikových systémech jako o další části ICS. Tento pohled není příliš definován z technologického hlediska, avšak z pohledu možné cesty vedení kybernetického útoku. Tuto hrozbu ve svých publikacích popsala řada autorů [10], [11], [12]. Tento pohled na problematiku sdílí i Idaho National Laboratory. Kim a Tran-Dang [12] ve své publikaci popisují Průmyslový internet věcí (IIoT) jako nový trend v rámci ICS. V souladu s tímto pojetím ICS systémů lze zavést tři základní pomyslné vrstvy: Podniková, Dohledová, Řídicí. Ty jsou dále

specifikovány pomocí jejich charakteristických prvků. Jednotlivé vrstvy jsou znázorněny v Obr. 1. Každá z vrstev reprezentuje hierarchickou úroveň řízení v architektuře ICS.



Obr. 1: Hierarchické vrstvy ICS [7]

2.1.1 Řídicí vrstva

Jedná se o nejnižší vrstvu v hierarchickém systému řízení ICS. Bezprostředně sleduje a reguluje řízené průmyslové procesy. K tomu využívá senzory a akční členy. Dále se v této vrstvě nachází Remote Terminal Unit (RTU) a Programmable Logic Controller (PLC). Ty jsou zodpovědné za řízení a kontrolu průmyslových procesů na základě svého programového vybavení. [7]

RTU

Remote Terminal Unit je elektronické zařízení, využívané pro řízení průmyslových procesů. Jeho funkce jsou podobné jako u PLC, avšak využití je odlišné. Jedná se o elektronické zařízení, které je neodmyslitelně spojeno se vzdáleným přenosem dat a odolnosti vůči přírodním vlivům. Právě proto je toto zařízení využíváno tam, kde není možno využít tradičního kabelového řešení. [7]

PLC

Programmable Logic Controller je průmyslový počítač, který zodpovídá za lokální řízení a kontrolu průmyslových procesů. To se odehrává na základě zpětné vazby ze senzorů, která je využita při nastavení akčních členů. PLC disponuje programovatelnou pamětí využívanou k ukládání instrukcí a funkcí. [7]

Akční členy a senzory

Tato elektronická zařízení jsou klasifikována jako nejnižší prvky v hierarchickém systému řízení ICS. Akční členy a senzory jsou zařízení, která ovlivňují a jsou ovlivňovány řízenými procesy. Sensory snímají kontrolované prostředí, o kterém informují kontroléry (PLC, RTU). Akční členy naproti tomu jsou vyrobeny jako zařízení využívající vstupy vytvořené kontroléry. Pomocí nich poté regulují průmyslové procesy. [7]

2.1.2 Dohledová vrstva

Tato vrstva je odpovědná za monitorování a dohled nad fyzikálními procesy. Také uchovává a dále využívá důležitá data, vztahující se k řízeným procesům. Do řízení systému je umožněno zasahovat obsluze prostřednictvím HMI. Operátoři využívají algoritmů, kterými upravují chování kontrolérů, čímž ovlivňují fyzikální procesy. [7]

Databáze historických dat

Jedná se o centralizovanou databázi historických dat týkajících se řízených procesů. S těmito uloženými daty lze provádět rozličné statistické analýzy a vyhodnocování. [7]

HMI

Jedná se o rozhraní mezi operátorem a řízeným systémem. V podstatě se jedná o základní prvek z pohledu řízení a kontroly uvnitř řízeného systému. Informuje operátory o aktuálním stavu řízených procesů. K tomu vyžaduje nezbytná data poskytovaná serverem pro sběr dat. HMI je také využíváno pro nastavení parametrů podle kterých kontroléry vykonávají svou činnost. [7]

Aplikační server

Aplikační server je odpovědný za řízení a směřování datového provozu v rámci jednotlivých aplikací. Přetváří data do správných formátů vhodných pro komunikaci. Také se stará o dodržování priorit při datové komunikaci. [7]

Databázový server

Jedná se o samostatný počítač, v němž je uložena příslušná databáze. Správce se stará o její správu. V tomto řízeném systému jsou poskytovány jednotlivé databázové služby vybraným aplikacím, které jsou důležité pro chod ICS. [7]

Server pro sběr dat

Server pro sběr dat poskytuje propojení služeb a Řídicí vrstvy (jedná se zejména o PLC a RTU). Přijatá data ze senzoru jsou posílána přes sběrnici do aplikací založených na internetovém protokolu (IP). K tomuto transferu je využito serveru pro sběr dat. To umožňuje uživateli dálkově přistupovat k datům o řízených procesech. [7]

MTU

Master Terminal Unit (MTU) neboli SCADA server je prvek, jenž ve SCADA systému vystupuje jako „Master“. Ostatní prvky jako RTU nebo PLC plní roli „Slave“. MTU přijímá data z RTU/PLC a dále je zpracovává podle potřeby. [7]

2.1.3 Podniková vrstva

Podniková vrstva je v rámci hierarchického uspořádání prezentovaných vrstev (Obr. 1) nejvyšší vrstvou. Účelem této vrstvy je především využití získaných dat pro účely samotného podniku, a to především jeho řízení, zvýšení zisku, zefektivnění marketingu a účetnictví, podporu administrativních a obchodních funkcí. Podniková vrstva je nejvíce ze všech vrstev závislá na konektivitě, a to především na nutnosti připojení k Internetu. Z tohoto důvodu je Podniková vrstva také považována za nejvíce otevřenou vrstvu. To se zásadně projevuje v rámci její využitelnosti pro průnik do dalších vrstev ICS. Podniková vrstva je první překážkou a cílem pro hackery k tomu, aby získali plnou kontrolu nad ICS. [7]

2.1.4 Dílčí závěr

Problematika klasifikace a rozčlenění ICS elementů je v dnešní době poměrně dobře stabilizována a popsána. Avšak jako všechny oblasti informačních a komunikačních technologií, tak i oblast ICS je ovlivněna kontinuálním vývojem. Autoři se shodují na základním rozdělení ICS, ve kterém reflektují základní skupiny produktů a logickou hierarchickou strukturu systémů ICS, která je do určité míry zobecněna. Je zde však fundamentální posun vnímání analyzovaných systémů od naprosto uzavřených systémů, za které byly ICS systémy v minulosti považovány, až po otevřené systémy, které jsou přístupny z kybernetického prostoru. Za určující důvod tohoto vývoje lze považovat stávající trend v propojování donedávna uzavřených ICS systémů s podnikovými systémy. Donedávna rozdílné oblasti systémů se v dnešní době čím dál více prolínají. Tato změna v probírané problematice vede k otázce, jestli by podniková vrstva měla být dále definována jako integrální součást ICS systémů se zásadním vlivem na ICS kybernetickou bezpečnost.

2.2 KYBERNETICKÁ BEZPEČNOST PRŮMYSLOVÝCH ŘÍDICÍCH SYSTÉMŮ

Na problematiku kybernetické bezpečnosti je často nahlíženo různými pohledy. Ty jsou v případě ICT a ICS do určité míry podobné, avšak jsou zde určité rozdíly. V rámci ICT a ICS systémů jsou sledovány především tyto oblasti priorit: dostupnost, důvěrnost a integrita dat. Jejich priority jsou znázorněny na Obr. 2 prostřednictvím zjednodušeného modelu. Z něho vyplývá zásadní závislost ICT na kritériu důvěrnosti. Naproti tomu je v rámci ICS považována oblast dostupnosti jako nejdůležitější. Tento rozdíl v prioritách zapříčiňuje vznik nových hrozeb pro ICS, které se v případě ICT nevyskytují, popřípadě nejsou tak

významné. Kontinuita provozu je z pohledu ICS tou nejchráněnější prioritou. Z tohoto důvodu je jakákoliv hrozba, ohrožující kontinuitu provozu, považována za kritickou. [8]

Obr. 2: Porovnání ICS a ICT ve vztahu ke kybernetické bezpečnosti. [8]

Tento zásadní rozdíl mezi ICS a ICT potvrdil autor Krotofil v knize [13], kde definoval tři základní rozdíly mezi ICS a ICT, které se opakují v rámci mnoha odborných publikací. V prvním bodu definoval kritéria dostupnosti a integrity jako fundamentálně významnější než kritérium důvěrnosti v případě ICS. Ve druhém bodu své publikace autor zdůraznil značnou závislost ICS systémů na plnění předdefinovaných procesů v čase. Z toho vyplývá potencionální újma při nerespektování této charakteristiky (např. sériově zapojený výpočetně náročný detektor anomálií). V rámci závěrečného bodu autor definoval ICS jako systém s poměrně dlouhým životním cyklem trvající až několik dekád. Tento fakt zapříčiňuje řadu problémů, které tvůrci těchto systému mohou jen těžce předvídat. Řada systémů ICS byla v minulosti navrhována z čistě provozního hlediska, kde kybernetická bezpečnost hrála jen okrajovou roli. Z tohoto důvodu je v rámci ICS systémů kybernetická bezpečnost řešena často nekoncepčním způsobem.

„*ICS má mnoho charakteristik, které se odlišují od tradičních IT technologií, včetně rozdílných rizik a priorit. Některé z nich představují významná rizika pro zdraví a bezpečí lidí, závažné škody na životním prostředí a finanční problémy, jako jsou výrobní ztráty a negativní dopad na národní ekonomiku. Oblast ICS má rozdílné požadavky na výkon a spolehlivost, a také na využití operačních systémů a aplikací, které mohou býti považovány za nekonvenční v tradičním IT prostředí. V souhrnu lze konstatovat, že provozní rozdíly a distribuce rizik mezi ICS a IT systémy vytváří požadavek na intenzivnější a sofistikovanější kybernetický-bezpečnostní a provozní strategie.*“ [7]

Oblasti hlavních rozdílů mezi kybernetickou bezpečností ICS a ICT byly definovány řadou autorů. V minulosti tyto rozdíly definovali autoři Luiijf a Paske [14], které podporuje publikace [7] od autora Stouffer. Toto rozdělení podporují i autoři Krotofil [13] a Patzer [15]. V nedávné době vznikla řada publikací, které pokrývají nově definovanou problematiku: Průmyslový internet věcí (IIoT). Tato

problematika vychází z aplikace metod IoT v oblasti ICS z důvodu zvýšení efektivity dosavadního řešení. Z tohoto pohledu je IIoT dalším logickým vývojovým prvkem v oblasti ICS, který ještě více umocňuje významnost kybernetických hrozeb v oblasti ICS. Hlavní rozdíly mezi kybernetickou bezpečností ICS a ICT jsou uvedeny v Tab. 2.

Tab. 2 - Porovnání kybernetické bezpečnosti v oblastech ICT a ICS [7], [13], [14], [15]

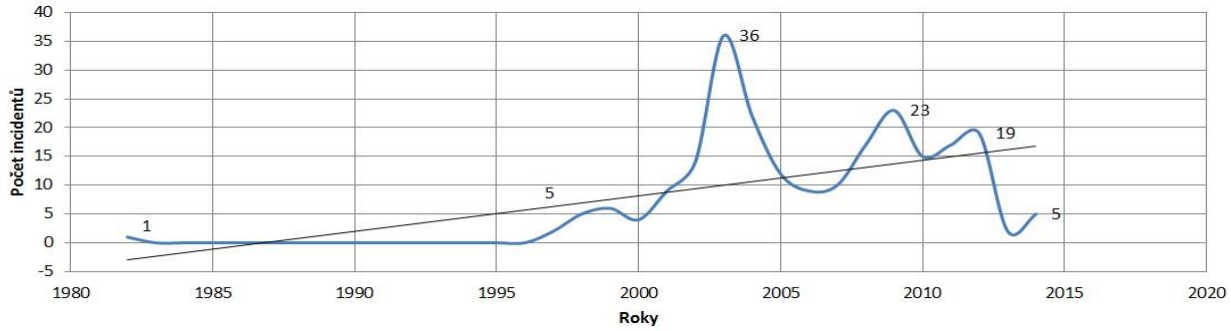
Porovnávané systémy /kritéria	***ICS***	***ICT***
Bezpečnostní priority	Dostupnost, viditelnost procesů, provozuschopnost procesů, integrita, důvěrnost	Důvěrnost, integrita, dostupnost
Dostupnost poskytovaných služeb	Kontinuální a nepřerušovaný provoz	Restart a přerušení provozu, když je potřeba
Latence	Požadavky na real-time provoz	Proměnlivé doby odezvy
Ochrana proti malware	Méně časté nasazení, nedostatečné zdroje pro aplikaci systémů pro ochranu před malwarem (zvláště u starších ICS)	Standardní
Patching	Aktualizace se považuje v rámci ICS za kritickou událost. Vyžaduje často schválení výrobci ICS popřípadě dodatečné certifikáty. Aplikace aktualizací je obtížná v kontinuálním provozu	Aktualizace jsou aplikovány ihned, když jsou dostupné
Hesla	Hard-coded hesla, která jsou přítomna v programovém prostředí	Pravidelně měněny
Životnost	15 – 30 let	3 – 5 let
Fyzická ochrana	Dobrá až výborná	Slabá (kancelářské systémy) až výborná (kritické systémy)
Vývoj bezpečnostních systémů	Není běžně integrální částí vývoje ICS	Integrální část vývoje ICT
Dopad rizik	Lidé, informace, produkce, životní prostředí	Informace
Komunikace	Mnoho proprietárních komunikačních protokolů, využití mnoha komunikačních prostředků	Standartní komunikační protokoly, z velké části kabelové komunikační prostředky

Stouffer [7] poukazuje na dosavadní historický vývoj, podle kterého je zřejmé, že valná část dosud využívaných ICS byla vyvinuta před soukromými a privátními sítěmi, jak je známe dnes, Internetem a personálními počítači. Tyto dnes již běžně využívané technologie jsou propojovány s ICS. Donedávna uzavřený systém se stává otevřeným. Tento vývoj vede ke tvorbě nových, dosud nepředpokládaných zranitelností.

2.2.1 Kybernetické incidenty týkající se průmyslových řídicích systémů

Z historického vývoje počtu kybernetických útoků na ICS, lze konstatovat, že v minulosti byly ICS systémy mnohokrát vystaveny kybernetickým útokům. V důsledku útoků došlo k únikům informací, destabilizaci systémů nebo narušení kontinuity provozu. V tomto kontextu je nutné definovat základní bod zlomu ve vztahu ke kybernetické bezpečnosti ICS. Takovým bodem bylo škodlivé působení počítačového červa Stuxnet v roce 2010. Ten měl charakter sofistikované kybernetické zbraně, která cílila na oblast ICS.

Z řady autorů zabývajících se výčtem i klasifikací kybernetických incidentů v oblasti ICS, můžeme jmenovat například Humayed [16] a Ani [17]. Avšak i přes značný zájem odborné komunity o řešenou problematiku, je většina studií zaměřena na poměrně malý výsek ICS kybernetických incidentů. Dosud největší a ucelený obraz o kybernetických incidentech poskytuje RISI databáze [18]. Tato databáze umožnuje kvantitativně analyzovat kybernetické incidenty pro období od 1982 do 2014. Bohužel, tato databáze není aktualizována od roku 2014. Obr. 3 popisuje vývoj počtu ICS kybernetických incidentů z dlouhodobého pohledu. Pomocí lineární regrese je možné vidět průběh trendu počtu kybernetických incidentů, který je v tomto případě vzrůstající. Lze usuzovat, že i nadále bude růst počet kybernetických incidentů zaměřených na ICS.[18]



Obr. 3: Historický vývoj kybernetických incidentů ICS [18]

Firma Kaspersky v roce 2018 [19] vydala studii, informující o meziročním zvýšení počtu útoků, vztahujících se k ICS systémům mezi roky 2017 a 2018. Napadené systémy disponovaly antivirem Kaspersky. Množství napadených ICS systémů ve druhém pololetí roků 2017 vyčíslila laboratoř „Kaspersky Lab ICS CERT“ na 37,75 % všech chráněných ICS systémů. V prvním pololeté roku 2018 došlo k navýšení celkového počtu napadených zařízení na 41,21 %. Tento trend

potvrdil v nedávné době rostoucí počet autorů publikujících v oblasti IIoT. Popisovaný vývoj byl zveřejněn v publikaci Tange [20]

2.2.2 Dílčí závěr

Kybernetická bezpečnost je v současnosti jednou z významných oblastí zájmu společnosti, přičemž lze predikovat růst její významnosti i v budoucnu. V rámci této podkapitoly byly analyzovány hlavní rozdíly mezi klasickou ICT kybernetickou bezpečností a ICS kybernetickou bezpečností. Hlavní rozdíl plyne z odlišného charakteru řešených oblastí. ICS systémy jsou často využívány jako součást kritické infrastruktury, pro kterou je zásadní prioritou poskytovat služby a prostředky pro společnost ve stanovené době a jakékoliv zpoždění nebo jejich výpadek lze považovat za kritický. Další zásadní rozdíl mezi vymezenými skupinami vyplývá z charakteru samotného ICS systému, který je vytvořený pro nepřetržitý dlouhodobý provoz (až 30 let). Z tohoto důvodu existuje řada ICS systémů vytvořených v době, ve které se s kybernetickou bezpečností v oblasti ICS ještě nepočítalo. Z toho vyplývají určitá omezení pro kybernetickou bezpečnost v této oblasti.

2.3 METODY DETEKCE KYBERNETICKÝCH ÚTOKŮ

Důležitost ochrany před kybernetickými útoky je primárně svázána s důležitostí chráněného aktiva. V tomto smyslu slova, lze konstatovat, že bezpečnost ICS systémů je jednou z nejkritičtějších oblastí bezpečnosti z důvodu kritické infrastruktury, ve které jsou ICS systémy využity. S příchodem čtvrté (digitální) průmyslové revoluce se řešená problematika kybernetické bezpečnosti stala ještě více významnou. Z tohoto důvodu je vhodné podrobit výzkumu detekční schopnosti a metody pro identifikaci kybernetických útoků. K detekci potencionálních kybernetických útoků je často využíván systém pro odhalování průniku (IDS), popřípadě systém pro prevenci před průnikem (IPS). Základní rozdíl mezi těmito systémy spočívá v přístupu k řešení incidentů. V případě IDS tuto funkci zastupuje vyškolený personál. Oproti tomu IPS jako proaktivní a prevenční systém autonomně a flexibilně reaguje na kybernetické útoky bez zásahu člověka.

Existují dvě hlavní skupiny detekčních metod pro kybernetické útoky. Toto rozdělení podporují publikace [7], [21]. První základní skupina detekce, založená na pravidlech, je také známa pod názvem detekce signatur (Signature detection). Do druhé skupiny lze zařadit oblast detekce anomálií, někdy také nazývanou jako detekce odlehlých hodnot nebo odchylek, která je určitým nástrojem pro oddělení normálního nezávadného chování od často výjimečného až anomálního chování, které často slouží jako příznak kybernetického útoku. Tato oblast detekce je do značné míry založena na metodách umělé inteligence, respektive na strojovém učení.

2.3.1 Detekce založená na pravidlech

Detekce založená na signaturách je základní metodou pro detekci kybernetických útoků, která je založena na komparaci. Proto jsou vytvořena pravidla, která jsou využita pro porovnání vzorů z reálného datového provozu s databází vzorů typických pro jednotlivé kybernetické útoky. Tato detekce je velmi efektivní vůči známým kybernetickým hrozbám, avšak je často neúčinná vůči dosud neznámým kybernetickým útokům nebo vůči modifikacím již známých kybernetických útoků. Každé pravidlo, založené na signaturách, začíná hlavičkou, ve které jsou uvedeny důležité informace pro její základní specifikaci. Dále je složeno z těla pravidla, které poskytuje doplňující informace pro identifikaci kybernetického útoku. Základní ukázka pravidla je zobrazena v Obr. 4.

```
alert tcp $EXTERNAL_NET 20000 -> $HOME_NET any (msg:"PROTOCOL-
SCADA DNP3 unsupported function code error";
flow:established,to_client; dnp3_ind:no_func_code_support;
reference:url,www.dnp.org/About/Default.aspx; classtype:protocol-
command-decode; sid:15718; rev:5;)
```

Obr. 4: Příklad pravidla využívaného v IDS Snort [22]

Alazab [23] ve své publikaci nastiňuje kombinaci detekce podle pravidel s detekcí anomálií. Navíc definují detekci podle pravidel jako:

„*SIDS (signature IDS) je založeno na technice párování vzorů za účelem nalezení známých útoků. Jinými slovy, když známé narušení odpovídá nebezpečnému řetězci v databázi, je aktivován poplach. SIDS obvykle poskytuje dobré detekční výsledky pro specifické, dobře známé útoky. Nicméně SIDS nemůže detekovat útoky nultého dne, z důvodu neexistence vzoru v databázi, dokud tam není nahrán. Hlavní výhodou SIDS je, že je velmi efektivní v detekci známých útoků bez vysokého počtu falešných poplachů.*“ [23]

2.3.2 Detekce anomálií

Detekce anomálií je progresivní metodou pro nalezení a oddělení vzorů, které se odchylují od tradičního chování. Chandola ve své publikaci [24] popisuje detekci založenou na anomáliích následovně:

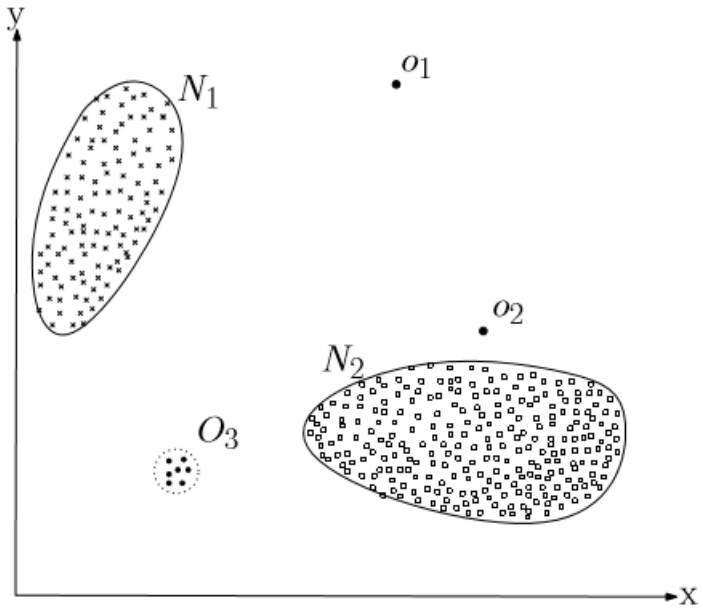
„*Detekce anomálií se týká problému nalézání vzorů v datech, které neodpovídají očekávanému chování. Tyto nevyhovující vzory jsou často označovány jako anomálie, odlehlé hodnoty, nesouhlasná pozorování, výjimky, odchylky, překvapení, zvláštnosti nebo kontaminanty v různých aplikačních oblastech.*“ Chandola [24]

Detekce anomálií není spojena jenom s kybernetickou bezpečností. Nalézá své uplatnění v celé řadě oblastí lidské činnosti, jako jsou například nejrůznější typy podvodů (finančnictví, telekomunikace, pojišťovnictví), chyby v nárocích na

zdravotní pojištění, neefektivita v účetnictví, e-mailový a webový spam, daňové úniky, monitorování aktivity zákazníků a profilování uživatelů, podvod s cennými papíry, nebezpečné nákladní zásilky, detekce malwaru / spywaru, falešná reklama, monitorování datových center, hrozba vnitřního nepřítele, obrazový a video dohled, detekce narušitele počítačové sítě a její selhání, detekce vesmírných objektů, detekce defektů na výrobcích, detekce chorob atd.

Anomálie jsou v rámci výpočetních systémů poměrně málo časté jevy, které se dají rozčlenit do dvou hlavních skupin. První skupinou jsou anomálie způsobené prostřednictvím úmyslné lidské činnosti, do které spadají kybernetické útoky. Druhou hlavní skupinou jsou anomálie, které vznikly působením neúmyslné lidské činnosti (např. špatná manipulace s kybernetickým systémem), popřípadě na základě přírodních poruch a chyb (např. šum), které jsou způsobeny prostřednictvím technické chyby, nedostatků technického vybavení nebo neúmyslného lidského působení.

Každá odchylka od normálního chování může být definována jako určitý příznak kybernetického útoku, který nemusí být do té doby známý. Existuje jen informace o změně v obvyklém chování systému. Z tohoto důvodu je **detekce anomálií** vhodnou detekční technikou pro identifikaci nových dosud neznámých nebo modifikovaných kybernetických útoků. Tyto vzory anomálií jsou často označovány jako odlehlé nebo extrémní hodnoty, které nejsou za normálních okolností přítomny ve sledovaném systému. Obr. 5 je zobrazen případ normálního neškodného provozu, který je reprezentovaný množinami N1, N2 a anomáliemi o1, o2 a kolektivní anomálií O3, které jsou definovány svým charakterem určujícím vybočující pozici oproti normálnímu provozu. Osy X a Y představují vybrané atributy ze získaného datasetu pro dvojrozměrný prostor.



Obr. 5: Anomálie ve dvojdimensionálním prostoru. [24]

Anomálie jsou vzhledem k charakteru použitých dat členěny do tří základních skupin bodová, kontextová a kolektivní anomálie. Ahmed [25] a v nedávné době Mirsky [26] definují tyto vymezené typy anomálií následovně:

- **Bodová anomálie** – Jedná se o typ anomálie „*kdy určitý datový objekt se liší od běžného vzoru v datasetu, lze jej nazvat jako bod anomálie. Například, když normální spotřeba paliva je definovaná jako pět litrů za den, ale pokud se změní na padesát litrů v jakémkoliv dalším dnu, tak poté se jedná o bodovou anomálii*.“ [25]

  Tento typ anomálie je znázorněn v Obr. 5, kde je reprezentován bodovými anomáliemi o1, o2.
- **Kontextuální anomálie** – Tento typ anomálie nastává „*když se datový objekt chová anomálně v souvislosti s konkrétním kontextem, lze jej nazvat jako kontextuální nebo podmíněnou anomálií. Například výdaje placené z kreditní karty během slavnostního období jako jsou Vánoce nebo Nový rok. Ačkoliv mohou být vysoké, nemusí být neobvyklé, neboť vysoké náklady jsou normální v době Vánoc. Na druhou stranu, stejně vysoké výdaje během měsíce bez jakýchkoliv slavností, lze považovat za kontextuální anomálii*.“ [25]
- **Kolektivní anomálie** – jedná se o skupinu datových objektů, které samy o sobě nejsou považovány za anomálie, avšak dohromady jako skupina prokazují chování anomálie ve vztahu k datům.

  „*Jako příklad uvažujme datový proud vytvořený vzorkováním paměti aplikace. Jediné pozorování z tohoto datového toku by mohlo patřit k libovolné aktivitě aplikace, a proto zkoumání její hodnoty samo o sobě nestačí k určení její abnormality. Spíše je třeba zvážit i její okolí*.“ [26]

Detekce anomálií s využitím umělé inteligence

Umělá inteligence je multidisciplinární vědní disciplínou, která je založena na imitaci procesů biologických, kde receptory jsou reprezentovány senzory a kognitivní funkce jsou reprezentovány algoritmem (modelem) jako jsou například umělé neuronové sítě. Podstatou umělé inteligence je přimět, naučit výpočetní techniku, aby vyřešila předložené problémy podobně jako by to udělal člověk. Jaký je však vztah mezi umělou inteligencí a strojovým učením? Fang [27] v publikaci definuje strojové učení jako podskupinu umělé inteligence, která se umí přizpůsobit novým okolnostem. [27] Z tohoto pohledu je strojové učení dynamické povahy s adaptivními schopnostmi. Algoritmy strojového učení jsou schopny adaptace a učení na nových datech bez jakéhokoliv zásahu člověka. Strojové učení využívá předložená data, ze kterých vyvozuje poznatky o jejich charakteru. Tyto znalosti umožňují vytváření predikcí o nových datech. Navíc, jakákoliv data, která nejsou zcela náhodná, v sobě obsahují vzory. Tyto znalosti poté algoritmus strojového učení generalizuje a vytváří prediktivní model kvůli oddělení důležitých informací od nepodstatných.

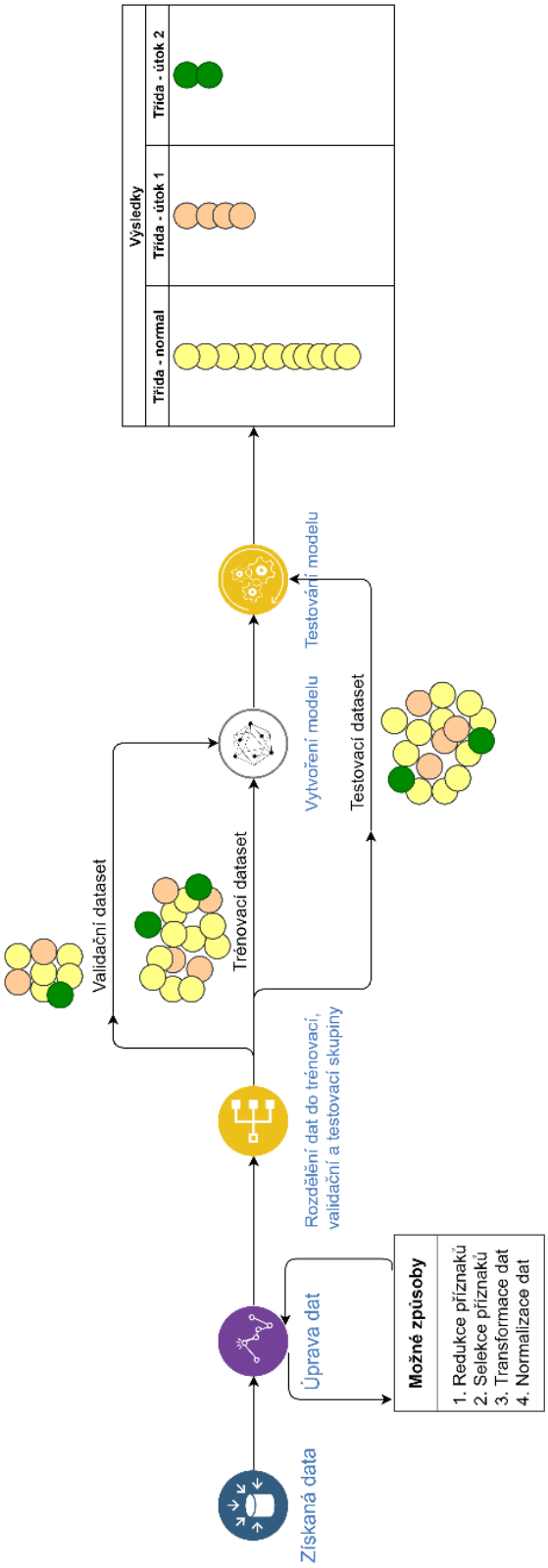
V dnešní době existuje řada algoritmů, které se mohou využít k detekci anomálií, pro klasifikaci anomálního a normálního chování a následnou separaci. Každý algoritmus pro detekci anomálií je nutné vybrat v závislosti na typu a charakteru dat. V závislosti na nich je vytvořen prediktivní model, který je využit pro další „nová“ data z důvodu identifikace anomálií. Následující podkapitoly se zabývají rozčleněním strojového učení do tří základních oblastí v závislosti na způsobu učení, predikce a formátu vstupních dat. Každý z algoritmů strojového učení lze klasifikovat do jedné z definovaných oblastí. Jedná se o oblasti: učení s učitelem (Supervised Anomaly Detection), učení bez učitele (Unsupervised Anomaly Detection), kombinace učení s učitelem a učení bez učitele (Semi-supervised Anomaly Detection). Tyto tři oblasti jsou podrobně definovány na příkladu detekce kybernetických útoků. U každé z těchto skupin se opakují postupy pro úpravy dat do podoby, která umožňuje jejich využití pomocí algoritmu pro strojové učení. Toto rozdělení podporuje množství autorů publikujících v oblasti detekce anomálií [21], [24], [28], [29], [30].

Učení s učitelem

Tato oblast detekce anomálií je založena na učení s učitelem. Mezi její hlavní nevýhody patří nutnost datasetu s označenými daty. Je zde nutná přesná specifikace anomálního a normálního provozu. Z tréninkových dat je pomocí klasifikačních algoritmů (klasifikátorů) vytvořen prediktivní model, který je evaluován na základě testovacích dat. Nevyváženost jednotlivých tříd v datasetu je v případě detekce anomálií častým problémem, který komplikuje klasifikaci dat. Tato problematika byla diskutována autorem Niu v publikaci [31]. Tento problém je zapříčiněn charakterem dat, ve kterém se anomálie vyskytují poměrně vzácně. Algoritmy, spadající do oblasti učení s učitelem, jsou elementárně závislé na formě vstupních dat neboli datasetu. Dataset je souborem získaných dat určitého formátu. Nejčastěji se jedná o maticovou strukturu, kde jednotlivé sloupce představují **atribut (proměnné, příznak)**, určité vlastnosti dat a řádky představují dílčí záznamy jednotlivých událostí. Je nezbytně nutné, aby každý datový bod (řádek v tabulce), byl označen, do jaké třídy patří. Upravená data jsou rozdělena do tří základních skupin (trénovací, validační a testovací dataset). Data pro trénování modelu jsou velmi důležitou částí celého procesu učení. Validační dataset je využit pro objektivní zhodnocení procesů spojených s trénováním, nastavení parametrů nebo zhodnocení generalizace modelu. Testovací dataset je vytvořen za účelem finálního zhodnocení vlastností vytvořeného modelu.

Na Obr. 6 je zobrazen proces trénování, validace a testování vytvořeného modelu prostřednictvím učení s učitelem. Je nutné poznamenat, že každý z datasetů nesmí sdílet společná data z důvodu zachování konzistentnosti validace a testování. Jelikož se jedná o tvorbu modelu učení s učitelem, tak každý z datových bodů je označen: žlutá barva představuje data v rámci normálního provozu sledovaného systému, růžová data náleží k útoku 1 a zelená data patří k útoku 2. V rámci této oblasti vytvořený model přesně klasifikuje testovací

dataset, jestli přijatá data spadají do oblasti normálního chodu systému, popřípadě jestli se jedná o útok typu 1 nebo útok 2. Nutno je však podotknout, že přístup k detekci anomálií zvláště v oblasti kybernetické bezpečnosti skýtá značné nedostatky. Zaprvé je velmi obtížné až nemožné získat dataset, který by obsahoval označená data všech kybernetických útoků. Tudíž jsou tyto algoritmy využity zejména pro systémy, v rámci, kterých je poměrně jednoduché získat dataset obsahující data pro všechny možné stavy. Nevýhody využití algoritmů založených na učení s učitelem diskutoval autor Rajendran [32], kde za hlavní nedostatek považuje uchovávaní a definici všech druhů anomálií, což je v řadě případů poměrně obtížné, popřípadě nemožné.



Obr. 6: Detekce anomálií založená na strojovém učení s učitelem. [vlastní zdroj]

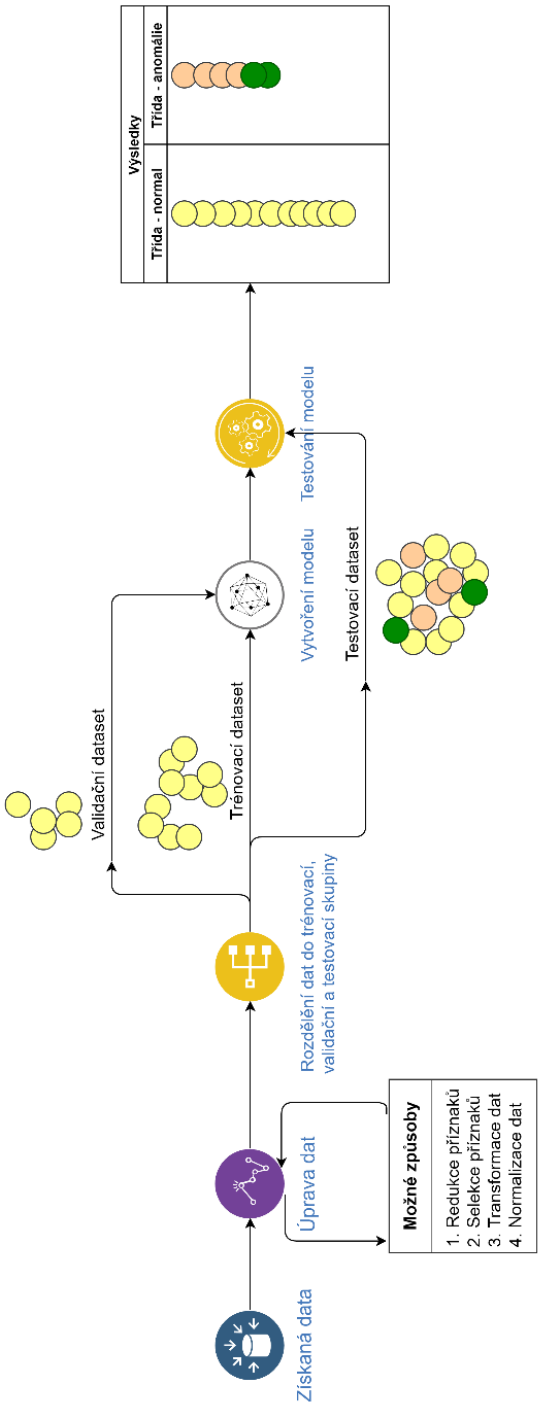
Kombinace učení s učitelem a učení bez učitele

Tato oblast detekce anomálií fundamentálně vychází z principu učení s učitelem, ale oproti učení s učitelem se zabývá pouze normálním a bezpečným provozem, který je obsažen v tréninkových datech. Anomální provoz je naopak obsažen pouze v testovacích datech. Problém nastává v případě, kdy tréninková data jsou nedostatečná nebo jsou kontaminována anomáliemi či projevy kybernetických útoků. Autoři Goldstein a Uchida [28] popisují oblast Semi-supervised Anomaly Detection.

„*Základní myšlenkou je vytvoření modelu normálních tříd, přičemž anomálie mohou být detekovány později tím, že se odchýlí od vytvořeného modelu. Tato myšlenka je také známa jako „one-class" klasifikace. Dobře známými algoritmy jsou „One-class SVMs" a „autoenkodéry*". " [28]

Tato skupina algoritmů strojového učení vychází z oblastí učení s učitelem a učení bez učitele. Podle Obr. 7 vypadá tato skupina algoritmu blíže učení s učitelem nežli učení bez učitele. Avšak jsou zde určité rozdíly. Vstupní data zastupují jenom jednu třídu. V oblasti kybernetické bezpečnosti jsou využívána data reprezentující normální a tím pádem i bezpečný provoz. Tato data jsou upravena a rozdělena do skupin (validační dataset, trénovací dataset a testovací dataset). Vytvořený model musí být „natrénovaný" a validovaný na datech, která se vztahují jen k jedné třídě. Tato data obsahují jen normální a bezproblémový provoz sledovaného systému bez přítomnosti anomálií (kybernetických útoků). Kybernetické útoky jsou obsaženy jen v rámci testovacího datasetu z důvodu evaluace vytvořeného modelu.

Výsledná data jsou poté rozdělena pomocí prediktivního modelu do dvou skupin. První skupina představuje data, která prediktivní model definuje jako normální provoz chráněného systému. Druhou skupinou je v podstatě vše ostatní. Tedy data, která se odlišují od dat využitých k trénování. Výhodou tohoto řešení je detekce neznámého chování ve sledovaných datech. Avšak tyto anomálie nelze dále specifikovat. Sledovaný systém detekuje určité vybočení z normálního provozu, avšak již nerozeznává, jestli se jedná o např. kybernetický útok nebo poruchu. V nedávné době řada autorů aplikovala algoritmy založené na kombinaci učení s učitelem a učení bez učitele na nejrůznější problémy vztahující se k detekci anomálií, [32], [33], [34] atd.



Obr. 7: Detekce anomálií založená na kombinaci strojového učení. [vlastní zdroj]

Učení bez učitele

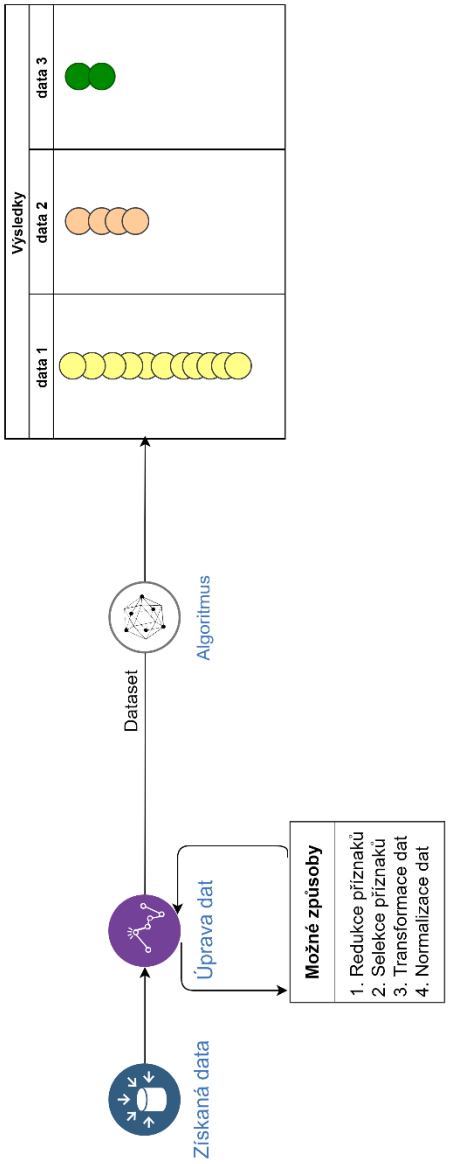
Jedná se o oblast, která nepotřebuje žádná trénovací data. Základní myšlenka této oblasti strojového učení je postavena na předpokladu v rámci, kterého je normální a neškodlivý provoz systému daleko více běžný než výskyt anomálií. Arunraj v publikaci [35] popisuje algoritmy založené na učení bez učitele následovně:

Techniky v této kategorii implicitně předpokládají, že normální případy jsou zastoupeny v testovacích datech mnohem častější než anomálie. Pokud tento předpoklad není pravdivý, pak tyto techniky trpí vysokým počtem falešných poplachů. [35]

Dataset je evaluován a vyhodnocen v závislosti na vzdálenosti a hustotě dat v prostoru. Tato oblast detekce anomálií je flexibilnější než ostatní oblasti detekce anomálií. Divya a Kumaran [36] definují základní ideu řešené oblasti detekce anomálií, přičemž vyzdvihují základní charakteristiku technik spadajících do této oblasti, kde je velmi nepraktické ověřování trénovacích dat. Ta nejsou potřeba v případě detekce anomálií založeném na učení bez učitele.

„*Techniky založené na učení bez učitele detekují anomálie bez znalosti třídy. Je předpokládáno, že anomálie jsou geometricky oddělitelné v n-rozměrném prostoru od normálního provozu.*“ [36]

Učení bez učitele není založeno na tvorbě modelu. Místo toho využívá algoritmu k tomu, aby vyhledal strukturu v rámci datasetu. Místo toho abychom našli odpověď na zadanou otázku, např. jestli určitá data pocházejí z kybernetického útoku, tak hledáme spojení mezi jednotlivými datovými body. Oproti učení s učitelem, zde nepotřebujeme žádné další informace o třídě. Také nevyužíváme modelů. Získaná data předložíme algoritmu, a ten je separuje do skupin v závislosti na charakteru dat. Výsledek může na první pohled vypadat podobně jako v případě učení s učitelem. Avšak je zde jeden podstatný rozdíl. Při vyšším počtu anomálií může dojít k falešné klasifikaci normálního provozu systému jako anomálie. Na tento nedostatek poukázal autor Arunraj [35] a také autor Saari [37]. Pro názornou demonstraci byl použít prezentovaný příklad viz. Obr. 8. Výhodou tohoto řešení je identifikace různých skupin dat bez další nutné znalosti a definice tříd.



Obr. 8: Detekce anomálií založená na strojovém učení bez učitele. [vlastní zdroj]

2.3.3 Aplikace metod strojového učení pro detekci kybernetických útoků v prostředí průmyslových řídicích systémů

Využití metod strojového učení pro ochranu ICS před kybernetickými útoky je poměrně nová a dynamická oblast kybernetické bezpečnosti. Tato specifická oblast bezpečnosti se stala předmětem zájmu po kybernetickém incidentu v Íránu (2010), kde počítačový červ Stuxnet destabilizoval jadernou elektrárnu v Búšehru. Tento milník vedl k zásadnímu přehodnocení kybernetické bezpečnosti v oblasti ICS. Uvedenou skutečnost začalo ve svých publikacích reflektovat značné množství autorů, kteří hledali nové způsoby ochrany ICS před kybernetickými útoky. V posledních letech se stále více autorů zaměřuje na kybernetickou ochranu ICS prostřednictvím technik pro detekci anomálií. Sokolov ve své publikaci [38] uvedl základní rozdělení, výhody a nevýhody

aplikace algoritmů strojového učení v prostředí ICS. Konstatoval nevhodnost použití systému detekce anomálií, založeném na lineárních modelech z důvodu nízké efektivity detekce anomálií. Dále vyzdvihl lepší detekční schopnosti skupiny technik založených na stromových strukturách (např. Random Forest). Výsledky však podle autorů ukazují na náchylnost k „přeučení“ vytvořeného modelu, a tudíž i horším generalizačním vlastnostem tohoto řešení. Poslední zkoumanou skupinou byly podle autorů neuronové sítě. Ty vykazovaly nejpřesnější výsledky, avšak za cenu vyšší výpočetní náročnosti, zvláště pak při velkém počtu vstupních dat. Liu se v publikaci [39] zaměřil na využití konvolučních sítí pro identifikaci významných atributů a anomálií v síťovém provozu ICS. Využil také algoritmu pro definici stavů sledovaného systému. Kombinací těchto řešení docílil robustnějšího řešení, které však mělo několik nedostatků, které bylo třeba odstranit. Jednalo se především o otázku interpretace, anebo zvýšení detekčních schopností navrhovaného řešení. Autoři Kravchik a Shabtai v jejich publikaci [40] aplikovali mnohovrstvou rekurentní neuronovou síť pro oblast detekce kybernetických útoků v rámci čističky odpadních vod. Přestože výsledky jejich výzkumu vykazují dobré výsledky, tak sami autoři konstatují limitované možnosti představeného řešení z různých pohledů, jako je například malá množina testovacích dat, malá interpretace výsledků, vybrané kybernetické útoky bylo poměrně lehké odhalit atd.

2.3.4 Dílčí závěr

Tato kapitola je zaměřena na oblast současných trendů v detekci kybernetických útoků. Samotné detekční techniky jsou základní a jednou z nejvýznamnějších částí systémů IDS. Ty budou v budoucnu hrát zásadní roli při ochraně ICS před kybernetickými útoky. Opačným případem je samotná oblast metod detekce kybernetických útoků. Každá z analyzovaných oblastí má své využití při detekci kybernetických útoků. Jejich výhody a nevýhody definují možnost využití každé detekční metody. První a základní je detekce založená na signaturách (pravidlech). Mezi její výhody patří jednoznačná identifikace kybernetických incidentů při nízkém počtu falešně identifikovaných kybernetických útocích. Tato oblast detekce má však zásadní slabinu ve způsobu činnosti. Každá signatura přesně odpovídá jednomu kybernetickému útoku, který musel být nejprve zaregistrován, analyzován a na základě toho mohla být vydána signatura. Ta byla distribuovaná do databází signatur, a proto je každý IDS systém účinný do té míry, do jaké má aktuální a kvalitní databázi signatur. Proces identifikace a analýzy kybernetických útoků je v mnoha případech značně náročný. Z tohoto důvodu je každý IDS, který je založen jenom na detekci pomocí signatur poměrně neúčinný vůči novým nebo málo známým kybernetickým útokům jako je např. APT.

Druhá perspektivnější oblast detekce kybernetických útoků je založena na identifikaci anomálií pomoci strojového učení. Podle dosavadního vývoje v této oblasti rozeznáváme tři základní sekce v řešené problematice dělené podle

charakteru vstupních dat a charakteru řešené úlohy. Do první oblasti spadají techniky založené na učení s učitelem, které vyžadují vyvážená trénovací data a také označení dat příslušící k normální třídě a dat příslušící ke kybernetickým útokům. Takto definovat data je velmi časově náročné, a navíc nepokrývají všechny kybernetické útoky. Oproti tomu pomyslné hybridní řešení ve formě kombinace učení s učitelem a učení bez učitele, popřípadě detekce založená na učení bez učitele mají dostatečný potenciál pro detekci neznámých kybernetických útoků, protože se zabývají chováním samotného systému, místo analyzování dílčích kybernetických útoků. Mnozí autoři popisují oblast detekce anomálií v rozsáhlém počtu publikací pro různé oblasti lidské činnosti. Avšak jen omezené procento z nich je zaměřeno na oblast kybernetické bezpečnosti pro ICS systémy. Značné množství autorů představuje slibná řešení, která však neberou v potaz aspekty a kritéria pro využití těchto detekčních systémů v prostředí ICS. Vyvstává proto řada otázek, na které je nutné najít dostatečnou odpověď. Jednou ze zásadních otázek pro nasazení metod strojového učení je výpočetní, a tudíž i časová náročnost detekčních modelů pro systém ICS. V řadě publikací není zohledněna problematika falešných poplachů a jejich minimalizace z důvodu ochrany dostupnosti služeb ICS a zajištění jeho kontinuálního chodu. Toto kritérium, jak již bylo popsáno v jedné z předešlých kapitol, představuje značné riziko pro systémy ICS. V neposlední řadě je zde otázka interpretace výsledků prediktivních modelů, které zasazuje každý detekční systém do potřebného kontextu a je tudíž nutné se touto problematikou zabývat. Základní nedostatky současného řešení a jejich implikace se dají shrnout do následujících bodů:

- nízká míra přesnosti v rámci identifikace kybernetických útoků – systém pro detekci anomálií je neúčinný v plnění své základní funkce (identifikace kybernetických útoků),
- vysoká míra falešných poplachů – vysoký počet falešných poplachů ohrožuje kontinuitu procesů v rámci ICS. Z tohoto pohledu může mít systém pro detekci anomálií negativní vliv na chráněný systém,
- vysoká výpočetní náročnost – vysoká výpočetní náročnost zvoleného řešení pro detekci anomálií způsobuje zvýšení zpoždění přenášené komunikace a procesů. Tento nedostatek řešení je zvláště kritický v případě ICS systémů,
- interpretace kybernetického útoku – identifikace anomálií sama o sobě nepřináší dostatek informací pro efektivní ochranu před kybernetickými útoky. Až interpretace zjištěné anomálie umožňuje rozpoznat, o jaké nebezpečí se jedná, a tím urychlit eliminaci vzniklé hrozby.

Na základě uvedených i jiných nedostatků současného stavu řešené problematiky jsem vyhodnotil nutnost koncepčního řešení problematiky. Vhodné a systematické řešení zvolené problematiky od úpravy datasetu až po optimalizace a interpretace navrženého řešení je nezbytnou cestou k bezpečné správě ICS systémů.

3. CÍLE DISERTAČNÍ PRÁCE

Hlavním cílem dizertační práce je:

Konceptuální návrh a ověření systému detekce anomálií z pohledu kybernetické bezpečnosti, založeného na strojovém učení, v průmyslových řídicích systémech.

K dosažení hlavního cíle bude nutné splnit tyto dílčí cíle:

- vymezení postupu identifikace kybernetických útoků pro systémy ICS,
- výběr, úprava a analýza vybraných ICS datasetů a jejich parametrů, které budou využity pro detekci anomálií,
- identifikace a analýza algoritmů strojového učení vhodných pro oblast detekce anomálií,
- využití optimalizačních technik pro zvýšení detekčních schopností zvoleného řešení,
- zhodnocení možnosti interpretace detekovaných anomálií,
- vytvoření algoritmu pro detekci anomálií, založeném na strojovém učení,
- ověřování, testování a hodnocení navrženého řešení.

4. ZVOLENÉ METODY ZPRACOVÁNÍ

Pro charakterizaci současného stavu v rámci řešené problematiky je využito řady publikací ve formě zahraničních odborných knih, mezinárodních konferenčních příspěvků, mezinárodních časopisových příspěvků a další odborné světové literatury. Pro úspěšné řešení cílů dizertační práce je využito následujících metod vědecké práce (metoda analýzy, metoda syntézy, metoda modelování, metoda komparace, metoda experimentu, metody matematické statistiky, metoda indukce).

- **Metoda analýzy** – jedná se o obecnou vědeckou metodu, která je založena na dekompozici zkoumaného jevu, na dílčí části, které jsou dále vyhodnoceny za účelem odhalení jejich podstaty. Analýza je založena na předpokladu, podle kterého lze definovat každý zkoumaný jev jako systém, který je složen z množiny prvků, které jsou spojeny jejími vlastnostmi a vztahy. Z tohoto pohledu metoda analýzy umožňuje oddělení podstatného od nepodstatného, směřuje od složitého k jednoduchému. Pro účely disertační práce je tato metoda využita k porozumění technik, dat a postupů, čehož bylo následně využito k jejich výběru.
- **Metoda syntézy** – tato vědecká metoda je založena na spojení dílčích částí do jednoho celku při sledování souvislostí, vazeb a vlastností mezi jednotlivými prvky jevu. Tento postup vede k hlubšímu pochopení zákonitostí fungování a vývoje jevu. V disertační práci je využita pro tvorbu systému detekce anomálií. K tomu je použito řady dílčích postupů, technik a algoritmů.
- **Metoda modelování** – metodu modelování můžeme v tomto případě definovat jako experimentální proces, jehož výsledkem je vytvoření abstraktního modelu, který je využit pro detekci komunikačních anomálií. Samotný model nabízí do určité míry zjednodušený obraz skutečnosti. V souvislosti s cíli dizertační práce je využito metody modelování k vytvoření prediktivního modelu pro klasifikaci zvoleného datasetu.
- **Metoda komparace –** jedná se o základní metodu pro porovnání dvou a více objektů v jednotném prostředí (stejné podmínky), popřípadě pro vyhodnocení dvou a více prostředí (rozdílné podmínky) pro jeden objekt. Na základě provedených experimentů lze následně vyhodnotit vlastnosti objektů. V rámci dizertační práce je použito metody komparace pro porovnání účinnosti jednotlivých metod detekce. K tomu je využito metod matematické statistiky.
- **Metoda experimentu** – tato empirická metoda je zaměřena na testování a ověřování pravdivosti vytvořených hypotéz za stanovených podmínek. Cílem je verifikovat neboli potvrdit nebo falzifikovat neboli vyvrátit platnost hypotézy. Metoda experimentu je jednou z důležitých metod

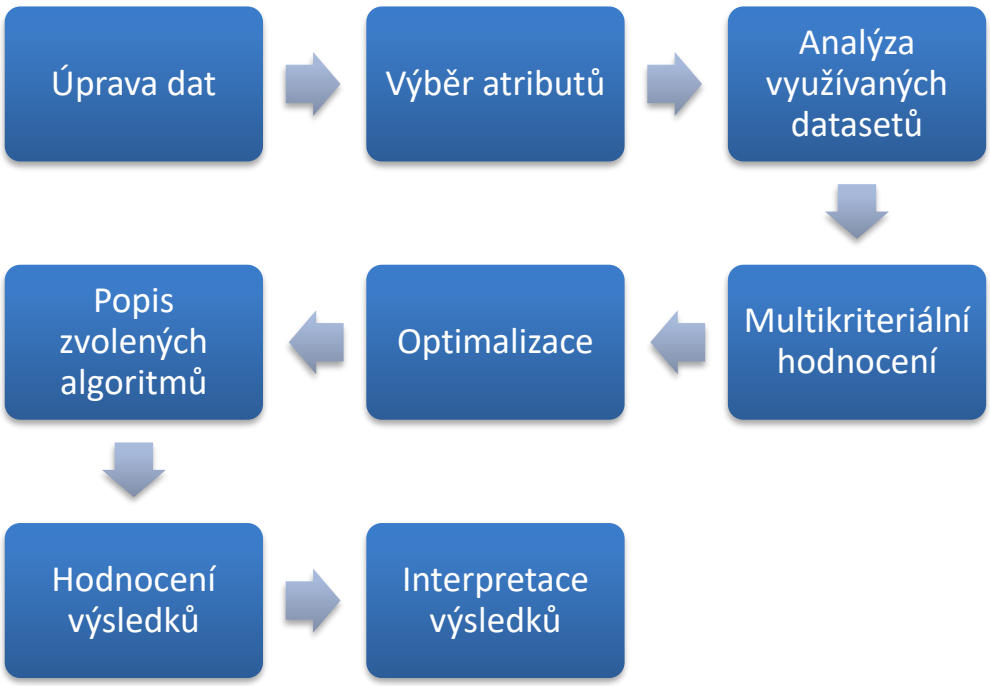
nutných pro naplnění cílů dizertační práce. Tato metoda je využita k ověření předpokladů v oblasti detekce anomálií.

- **Metody matematické statistiky** – jedná se o exaktní disciplínu, která je založena na analýze empirických dat. Využití těchto metod umožňuje analyzovat data za účelem přesné specifikace jevů a jejich vztahů. Metody matematické statistiky jsou v rámci dizertační práce využity pro analýzu datových setů a v oblasti vývoje vhodných postupů při identifikaci anomálií.
- **Metoda indukce** – tato vědecká metoda je založena na generalizaci sledovaných jevů. Jedná se o nepříliš přesnou metodu, která ve svém důsledku může vést k zavádějícím předpokladům. Z tohoto důvodu je nutné využit dostatečný počet různorodých pozorování sledovaného jevu. Výsledek indukce vypovídá o podstatě zákonitostí ve sledovaném jevu. Výstupem této vědecké metody bývá zpravidla hypotéza, která je buď potvrzena, nebo vyvrácena. Tato vědecká metoda je využita ke tvorbě hypotéz v rámci oblasti detekce anomálií.

5. TEORETICKÝ RÁMEC

Účelem této kapitoly je vytvořit teoretický základ použitých technik a algoritmů za účelem objasnění zvolených postupů využitých pro naplnění vytyčených cílů disertační práce. Tato struktura teoretického rámce vychází ze základního dělení procesu aplikace algoritmů strojového učení. Metody a algoritmy strojového učení tvoří jádro disertační práce, která je zaměřena na automatickou identifikaci anomálií v prostředí ICS. Právě strojové učení bylo v rámci disertační práce identifikováno jako velmi vhodná oblast řešení pro oblast detekce anomálií.

Navržený algoritmus vychází z osmi základních částí (zobrazeny v Obr. 9). Naplnění těchto částí je nutným předpokladem ke splnění dílčích cílů disertační práce. První část teoretického rámce disertační práce je zaměřena na popis procesů, které je nezbytné aplikovat před aplikací algoritmů pro strojové učení. Obsahem této části je objasnění nutných úprav získaných datasetů pro algoritmy strojového učení. Právě úpravy (transformace) datasetu jsou základní složkou aplikace algoritmů strojového učení, která zásadně ovlivňuje jejich výkon.



Obr. 9: Kapitoly teoretického rámce disertační práce. [vlastní zdroj]

Ve druhé části teoretického rámce je rozebrána oblast výběru atributů, kde jsou popsány metody výběru atributů a redukce atributů využité v rámci disertační práce. Ve třetí části teoretického rámce jsou identifikovány a analyzovány datasety, které jsou využity pro trénování a evaluace prediktivních modelů. Ve čtvrté části teoretického rámce jsou popsány využité algoritmy strojového učení. Ty tvoří jádro disertační práce, které je zaměřeno na detekci anomálií. Jejich popis je tudíž nezbytným krokem k naplnění všech cílů. Optimalizační techniky

jsou popsány v páté části. Ty jsou využity pro nastavení hyperparametrů algoritmů strojového učení pro zvýšení jejich detekčních schopností. Zvláště pak ve spojení s multikriteriálním hodnocením (šestá část) jsou optimalizační techniky využity k nastavení každého z dílčích algoritmů strojového učení nejenom podle přesnosti detekce, ale také podle dalších kritérií jako je např. počet falešných poplachů nebo výpočetní náročnost algoritmu.

Metodika způsobu hodnocení výsledků detekčních vlastností dílčích prediktivních modelů je popsána v části sedm. Závěrečná část teoretického rámce disertační práce je zaměřena na popis využitých technik pro interpretaci výsledků. Ty jsou využity pro interpretaci výsledků v rámci detekovaných anomálií.

5.1 ÚPRAVA DAT

Úprava a výběr vstupních dat je jednou z velmi důležitých částí pro efektivní využití algoritmu strojového učení. Často však jsou tato data ukládána v rozdílných formátech. Proto je nutná jejich transformace do podoby, která vyhovuje příslušným algoritmům. Je vhodné poznamenat, že obecně strojové učení využívá numerické hodnoty pro trénování modelů. Proměnné (neboli atributy) reprezentují jednotlivé vlastnosti zkoumaného jevu. V tomto případě je můžeme definovat jako sloupce v pomyslné dvojdimensionální tabulce, kterou algoritmy pro strojové učení využívají k vytvoření modelů. Rozeznáváme několik druhů datových reprezentací v závislosti, na kterých využíváme rozdílné postupy pro jejich transformaci. Podle typu je nutné transformovat data do vhodné podoby, v tomto případě do číselné podoby.

5.1.1 Dělení druhů dat a jejich transformace

V této podkapitole jsou definovány základní typy datových reprezentací. Jedná se o nominální data, ordinální data, intervalová data a poměrová data. Tato množina dat představuje souhrn základních typů datových reprezentací, které byly využity v rámci disertační práce.

Nominální data

První druh dat označujeme jako nominální data a můžeme je začlenit do oblasti kvalitativní reprezentace dat. Nominální reprezentace dat může nabývat dvou a více hodnot, které se vzájemně vylučují a nedají se mezi sebou porovnat v rámci velikosti. Příkladem může být definice pohlaví člověka (muž, žena). Porovnání velikosti zástupců v této binární reprezentaci nepřináší nové informace, a proto nemá smysl. Transformace těchto dat do numerické podoby je velmi složité. Jeden ze způsobů transformace je zmapování všech zástupců v datasetu a následné přidělení numerické hodnoty odpovídající určité skupině zástupců (např. v případě barev: červená – 1, modrá – 2, zelená – 3, hnědá – 4, fialová – 5 atd.). Takto transformovaná data jsou v určitém směru funkční pro potřeby

strojového učení, avšak nejsou optimální, jelikož touto transformací jsou zavedeny vztahy mezi jednotlivými zástupci, které před dotyčnou transformací nebyly. Například modrá barva je nejméně významná, zatímco fialová barva je nejvíce významná, nebo fialová barva je dvakrát významnější jako zelená atd. Vhodnější je využít one-hot encoder (OHE), který nominální data transformuje do binární podoby. Proces transformace zahrnuje převod unikátních hodnot v rámci jedné proměnné na reprezentaci binárního vektoru, který nabývá dimenze podle počtu unikátních hodnot. Tento postup má však i své úskalí. V rámci velkého počtu unikátních hodnot dochází k tzv. prokletí dimensionality, tedy k vytvoření velmi rozsáhlých datasetů, což může vést ke zkreslení výsledků a velké výpočetní náročnosti.

Ordinální data

Druhou skupinou dat jsou ordinální data. Stejně jako u nominálních dat, jsou tato data reprezentována kategoriemi, které jsou vyjádřeny znaky, avšak jednotliví zástupci mají mezi sebou jasně definované vztahy. Příkladem ordinálních dat může být například dotazníkové šetření, kde jednotliví účastníci zodpovídají otázku, jak moc byli spokojení se zakoupeným produktem. Přičemž dotazovaný má k dispozici výběr ze čtyř odpovědí (nízká spokojenost, mírná spokojenost, spokojenost, naprostá spokojenost). Každá z těchto odpovědí má vztah k ostatním a je zde patrný vzestupný trend, který lze vyjádřit pomocí vzestupné číselné reprezentace, například pomocí Likertovy škály. Tato škálovací metoda se hojně využívá u konverze dotazníkových šetření do numerické podoby, jelikož data v rámci dotazníku často nabývají ordinálního charakteru. Jedná se o přiřazení celého čísla v závislosti na významnosti dat.

Intervalová data

Intervalová datová reprezentace spadá do skupiny kvantitativních dat. Jedná se tedy o hodnoty, které jsou vyjádřeny numericky, přičemž každá jedna hodnota je stejně vzdálená od předešlé. Většinou se jedná o hodnoty měřené v rámci stupnice jako je čas nebo teplota. Výhodou těchto dat je možnost jejich srovnání. Rozdíl mezi časem 14:00 a 15:00 je hodina, která má stejnou velikost jako hodina získaná rozdílem časů 20:00 a 21:00. Problémem těchto dat je absence nuly, což má určité konsekvence v oblasti matematické operace. V podstatě lze s daty pracovat bez další transformace, jelikož již jsou v číselné podobě. Lze využít techniky „data binning“ pro snížení počtu rozsahů. Tato technika je podobná kvantování v případě signálů, kde data spadající do určitého intervalu, jsou nahrazena jednou hodnotou.

Poměrová data

Poslední hlavní skupinou dat jsou poměrová data. Tato numerická data mají jasně definované vztahy mezi jednotlivými hodnotami. V podstatě do této kategorie spadají všechny SI veličiny, jako je například rychlost, hmotnost,

elektrický proud atd. Na rozdíl od intervalových dat mají poměrová data jasně definovanou nulu. V jiných ohledech mají v zásadě stejné vlastnosti jako intervalová data. Jak již bylo řečeno, tak popisovaná data jsou definována jako numerická. Proto většinou není potřeba dalších transformací pro jejich další využití pro algoritmy strojového učení.

5.1.2 Chybějící hodnoty

Řada systémů při svém provozu generuje množství tzv. „chybějících hodnot“, které mohou být problémem pro řadu algoritmů strojového učení. Chybějící hodnoty mohou vznikat chybami v měření, nebo se běžně objevují v závislosti na zvoleném způsobu měření a tvorby dat. V případě nominálních dat je řešení jednoduché z pohledu implementace. Výsledné chybějící hodnoty jsou uvažovány jako další kategorie nominálních dat. V případě numerických dat je však situace komplikovanější. Základní metodikou postupu by v nejjednodušším případě při výskytu nulové hodnoty bylo vymazání celého datového bodu (záznamu). Tento postup by však při větším výskytu nulových hodnot znamenal značnou redukci velikosti datasetu, jenž by vedla ke ztrátě důležitých informací. Tato technika se využívá jen v omezené míře při velmi malém počtu chybějících hodnot. Často chybně využívaným postupem je nahrazení všech chybějících hodnot jednou statickou hodnotou např. nulou. Tento postup předpokládá výskyt vybrané hodnoty ve všech atributech. Z tohoto důvodu je ovlivněna distribuce dat a tím pádem i celkový výsledek. Více využívaná varianta je nahrazení dotyčného pole novou hodnotou, která je aritmetickým průměrem nebo mediánem hodnot z každého individuálního atributu. Uvedený postup reflektuje více charakter dotyčného atributu, rozložení dat, a tudíž neovlivňuje výsledek v takové míře jako předchozí řešení. Důležitým předpokladem pro využití navrhovaného řešení je zachování pomyslné oddělenosti výpočtu pro chybějící hodnoty mezi trénovacím a testovacím datasetem.

5.1.3 Normalizace atributů

Normalizace jednotlivých atributů je dalším nutným krokem ke tvorbě prediktivního modelu. Prvním důvodem k normalizaci datasetu je rozdílné měřítko mezi jednotlivými atributy. Tyto proměnné se od sebe mohou zásadně lišit (např. až v řádech tisíců). Nabývá-li atribut 1 diametrálně odlišných hodnot (např. větší v rámci řádů), než atribut 2 poté atribut 1 bude mít větší váhu oproti atributu 2 v závislosti na rozdílném rozsahu hodnot. Také řada algoritmů strojového učení je v určité míře založena na výpočtech euklidovské vzdálenosti mezi jednotlivými datovými body. Pro provedení této transformace lze využít vztah (1), kde $\mathbf{min_v}$ představuje minimální hodnotu z množiny hodnot pro atribut a $\mathbf{max_v}$ představuje maximální hodnotu z této množiny. Proměnné $\mathbf{nová\_max_v}$ a $\mathbf{nová\_min_v}$ představují maximální a minimální hodnoty nového rozpětí (často využívané rozpětí je 0 a 1) a proměnná **x** představuje transformovanou hodnotu, kde $x \in \mathbb{R}$.

$$x^* = \frac{x - min_v}{max_v - min_v}(nová\_max_v - nová\_min_v) + nová\_min_v \quad (1)$$

Podobně jako v předchozím případě pro výpočet chybějících hodnot, tak i v tomto případě je nutno normalizaci provést v rámci trénovacího datasetu, tedy i získání nové škály pro transformaci nových numerických dat získaných v rámci testovacího datasetu, které mohou být diametrálně odlišné.

Standardizace je další technikou použitou pro změnu měřítka využívaných dat. Cílem této metody je vytvoření datasetu, který bude mít nulovou střední hodnotu a jednotkový rozptyl. Distribuce datasetu je poté centralizována v okolí nuly.

$$x^* = \frac{x - \mu}{\delta} \quad (2)$$

Kde $\boldsymbol{\mu}$ je střední hodnota a $\boldsymbol{\delta}$ představuje směrodatnou odchylku.

5.2 VÝBĚR ATRIBUTŮ

Oblast „výběr atributů“ také známá jako „výběr rysů“, je základním úkonem k vytvoření efektivního modelu v oblasti strojového učení. Jedná se o výběr nejvhodnější podmnožiny dat z celého datasetu, který může obsahovat stovky až tisíce atributů. Hlavní myšlenou této oblasti je tvorba redukovaného datasetu při zachování jeho informační hodnoty. Cílem technik pro výběr atributů je minimalizování výpočetního výkonu pro vytvoření a využití prediktivního modelu za cenu omezené ztráty informací.

Existují dvě základní skupiny technik pro výběr atributů. První skupinu technik je možné souhrnně nazvat selekcí atributů. Tato skupina technik využívá statistických metod pro výpočet významnosti každého dílčího atributu. Na základě čehož je poté vybrána množina nejvýznamnějších atributů. Základní souhrn technik pro výběr atributů popsali ve své publikaci autoři Miao a Niu [41]. V této publikaci je nastíněna řada technik pro výběr atributů jako například techniky založené na korelaci, při které jsou vyloučeny vysoce korelované, tudíž lineárně závislé atributy. „Wrapper“ (obalovací) techniky jsou založeny na využití algoritmu strojového učení pro proměnné množiny atributů datasetu, které jsou vyhodnoceny podle výsledku použitého algoritmu. Výsledkem tohoto postupu je poté nalezení podskupiny atributů, které nejvíce přispívají k dobrým výstupům zvoleného algoritmu. Tento postup se však příliš nedoporučuje, díky své povaze procházet značné množství možností a z toho vyplývají značné nároky na výpočetní zařízení. Možnosti pro eliminaci tohoto nedostatku umožňují v poslední době více využívané genetické algoritmy, které díky své podstatě procházejí podstatně menší prostor i v rámci velmi velkých datasetů. Existuje řada dalších technik pro selekci atributů, avšak z povahy této skupiny technik vyplývají nedostatky, které vyvstávají v případě systému pro detekci anomálií. Z podstaty detekce anomálií, tedy úkolu binární klasifikace, vyplývají základní charakteristiky, které znemožňují aplikaci definovaných technik. Řada

z postupů předpokládá obdobné chování v případě trénovacího a testovacího datasetu, avšak tento předpoklad v případě problematiky detekce anomálií neplatí. Naopak tyto dva datasety mohou být často rozdílné již z jejich povahy, kdy v trénovacím datasetu nejsou přítomny anomálie, zatímco v testovacím datasetu anomálie přítomny jsou. Další část technik předpokládá existenci tříd v datasetu, podle kterých mohou vyhodnotit významnost jednotlivých navržených řešení. To však v případě detekce anomálií je jen velmi obtížně dosažitelné. Tento způsob řešení by poté mohl zkonvergovat do jednoho výstupu, který vyhovuje jen specifickým případům. Z těchto důvodu je vhodné využít technik, spadajících do druhé skupiny, která se souhrnně nazývá „redukce atributů“. Techniky, spadající do této oblasti, obsahují postupy pro vytvoření nových atributů, založených na kombinaci předešlých. Tímto postupem je zajištěna redukce dimenze datasetu při zachování velké většiny informační hodnoty původního datasetu.

5.2.1 Analýza hlavních komponent

Z důvodu redukce dimenze využívaných datasetů bylo nutné vybrat řešení, které efektivně sníží dimenzi datasetu a umožní interpretaci výsledků. Jako vhodné řešení tohoto problému byla vybrána metoda analýzy hlavních komponent (Principal Component Analysis – PCA). Tato metoda je založena na kovariančních maticích, které vyjadřují vzájemnou závislost mezi popisovanými proměnnými a jejich směrodatnými odchylkami. Základní myšlenkou PCA je redukovat počet atributů, a přitom zachovat jejich původní informační hodnotu, tak aby nově vzniklé atributy obsahovaly vyšší variabilitu než původní atributy. Prakticky jsou původní data projektována do nižší dimenze. To lze za předpokladu existence hlavních komponent. Každá z nich reprezentuje množinu původních proměnných za předpokladu zachování variability. Nově vytvořené proměnné jsou nazývány hlavními komponentami v rámci, kterých je obsažena původní informační hodnota datasetu. Základem PCA je definice dvou funkcí, které umožní transformovat data z prostoru $\mathbf{R^k \rightarrow R^l}$, kde můžeme definovat vztah $\mathbf{R^l < R^k}$. Tyto funkce lze definovat podle vztahu (3) pro kódování a vztahu (4) pro dekódování. [42]

$$Y = f(X) \tag{3}$$

$$X = h(Y) \tag{4}$$

Kde kódovací funkce transformuje data **X** do podoby **Y** a dekódovací funkce provádí tento proces inverzně. Transformace datasetu začíná standardizací dat podle vztahu (2). Je stanovena kovarianční matice $\mathbf{M_k}$ pro definování vztahu mezi atributy v rámci jednoho datového bodu. Pro tři atributy (a, b, c) v rámci jednoho bodu platí vztah (5) pro jednotlivé kovariance. [42]

$$M_k = \begin{pmatrix} K(a,a) & K(a,b) & K(a,c) \\ K(b,a) & K(b,b) & K(b,c) \\ K(c,a) & K(c,b) & K(c,c) \end{pmatrix} \quad (5)$$

Další krok zahrnuje výpočet matice vlastních vektorů **E**, vycházející z $\mathbf{M_k}$. V rámci matice **E** jsou jednotlivé vlastní hodnoty seřazeny sestupně. Nový dataset **Y** disponující redukovanou dimenzí, kterou určuje počet hlavních komponent. Tento nový dataset vznikne vynásobením standardizovaného datasetu $\mathbf{X_s}$ s maticí vlastních vektorů **E**. [42]

5.3 ANALÝZA VYUŽÍVANÝCH DATASETŮ

V rámci této kapitoly jsou podrobně analyzovány datasety, které byly využity pro tvorbu a testování systému detekce anomálií prostřednictvím algoritmů strojového učení.

5.3.1 ICS Modbus dataset

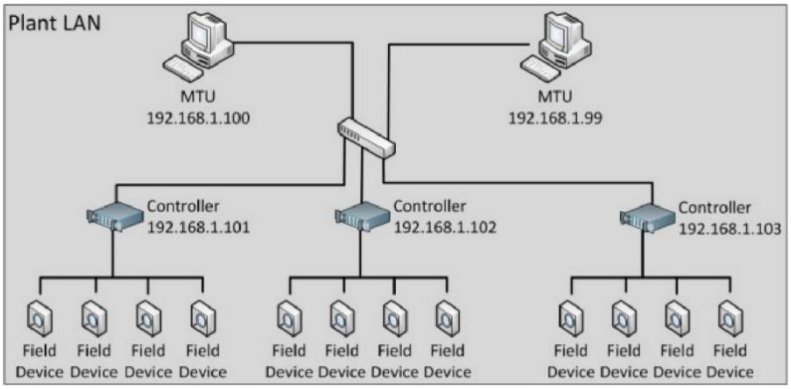
Z důvodu sestavení systému pro detekci anomálií bylo nutné využít dataset, tedy záznam komunikace v rámci ICS systému. Proto bylo využito ICS datasetu, který využívá Modbus komunikační protokol. Tento dataset byl prezentován v publikaci „Providing {SCADA} Network Data Sets for Intrusion Detection Research“. [43] Pro přehlednost je tento dataset v disertační práci dále pojmenován jako **dataset 1**.

Dataset sestává z několikahodinového záznamu síťové komunikace ICS systému, ve kterém jsou označeny kybernetické útoky spolu s normálním síťovým provozem. Data v rámci datasetu jsou ve formátu pcap, tudíž je nutné extrahovat vybrané parametry síťové komunikace do formátu csv, tedy do formátu dvojrozměrné tabulky s vybranými atributy. Data byla vygenerována pomocí řady simulací. Výsledný systém je zobrazen v Obr. 10 a představuje elektrickou distribuční síť se zdrojem o 12 000 V. Tento systém je složen z jedné hlavní větvě a tří menších větví, které jsou opatřeny vypínači. Vytvořený SCADA sandbox je složen z několika MTU a RTU, které komunikují prostřednictvím Modbus komunikačního protokolu. [43] Využívané atributy síťového provozu, extrahované z pcap souborů, jsou prezentovány v Tab. 3. Tyto atributy nemají v tomto případě strukturu, a proto jsou uvedeny v tabulce bez závislosti na řádku a sloupci tabulky.

Tab. 3 – Využité atributy síťového provozu v rámci datasetu 1. [43]

Síťové atributy							
Zdrojová IP	Cílová IP	Protokol	Zdrojový port	Cílový port	Flag	Flags - syn	Flags - push
Flags - ack	Flags - fin	Mbtcp - identi. jednotky	Mbtcp - identi. protokolu	Modbus - bit číslo	Modbus - data	Modbus - bit počet	Modbus - bit hodnota
Modbus - byte počet	Modbus - kód funkce	Modbus - padding	Modbus - referenční číslo	Modbus - číslo registru	Modbus - hodnota registru	Modbus - čitač slov (registrů)	Checksum - status
Propojení dokončeno	Propojení potvrzeno	Propojení - syn	Flags - cwr	Flags - řetězec	Tcp - option_kind	Tcp -options sack_perm	Delka rámce
Délka IP	Velikost okna	Počet bitů poslaných po PSH flag	Délka segmentu TCP	Velikost hlavičky	Čas delta	Modbus - zbývající délka paketu	TCP option - délka
TCP option - MSS velikost	PDU - velikost	Relativní čas	TCP - následující číslo sekvence	Velikost okna - faktor škálování	Časové razítko - “echo reply”	TCP - číslo sekvence	

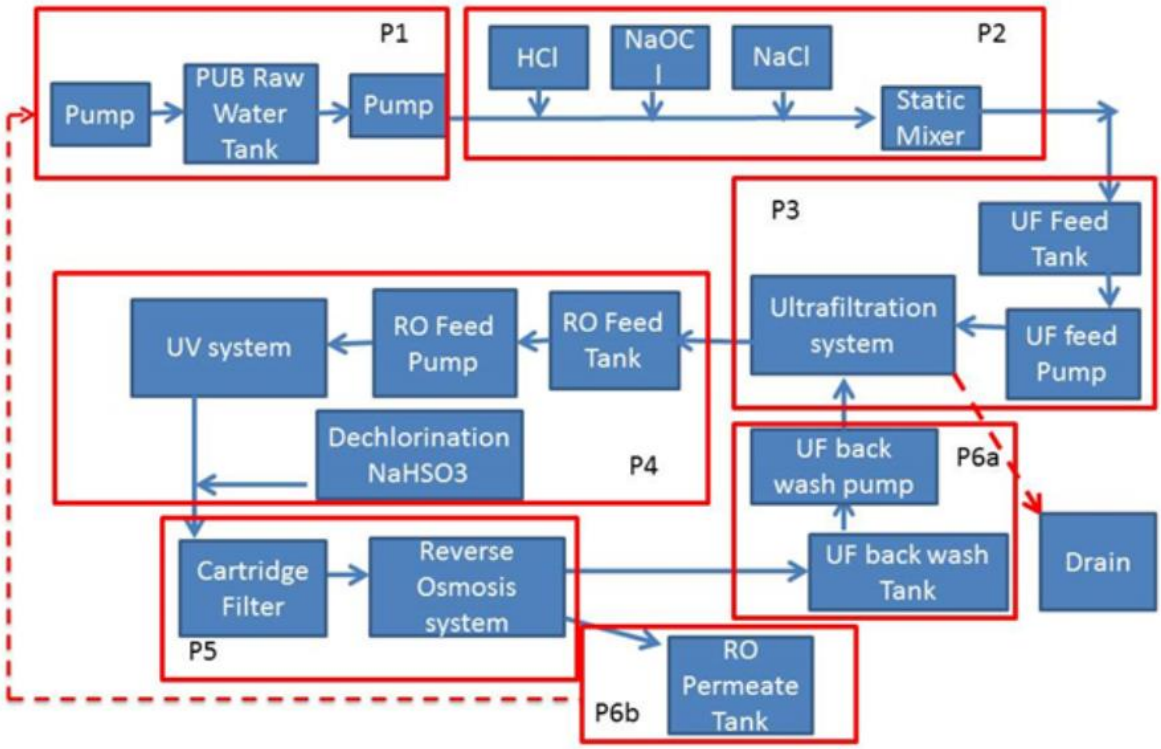
Pro generování kybernetických útoků byl využit nástroj Metasploit. Pro experimenty byly vybrány čtyři následující kybernetické útoky. Pro přehlednost jsou označeny od **CA1_1** do **CA1_4**. První kybernetický útok je zaměřen na distribuci „exe“ souborů z napadeného RTU v rámci jedné sítě (**CA1_1**). Druhý kybernetický útok je založen na zneužití exploitu (ms08_netapi), který je využit k šíření a nakažení RTU ve sledované síti (**CA1_2**). V rámci třetího kybernetického útoku byla provedena manipulace (přesunutí) souborů v rámci chráněného systému. Tento útok odpovídá konfiguraci napadeného systému útočníkem z důvodu aktualizace malware v rámci napadeného systému (**CA1_3**). Poslední kybernetický útok představuje zneužití nepovoleného příkazu, který je zaslán kontroléru (**CA1_4**). [43] Tyto kybernetické útoky byly využity v kapitole (6.2).



Obr. 10: Využitý ICS systém. [43]

5.3.2 Čistička odpadních vod řízená pomocí ICS (SWaT)

Jedná se o reálně postavený ICS systém, v rámci, kterého je prováděn výzkum v oblasti kybernetické bezpečnosti na univerzitě: „University of Technology and Design“ v Singapuru. [44] Pro přehlednost je tento dataset v disertační práci dále pojmenován jako **dataset 2**. Tento systém produkuje přibližně 20 l filtrované vody za minutu. Systém čištění vody sestává ze šesti základních procesů (viz. Obr. 11). V rámci prvního procesu (P1) je znečištěná voda přemístěna do vodovodní nádrže. V druhém procesu (P2) je zkontrolována míra znečištění vody ve vodovodní nádrži. Pokud kvalita vody vybočuje z nastavených norem, tak jsou aplikovány potřebné chemikálie. V rámci třetího procesu (P3) je využito filtračních membrán a ultrafiltrace pro odstranění částic znečištění a mikroorganismů. Čtvrtý proces (P4) je zaměřen na odstranění přebývajícího chloru z vody pomocí dechloračního procesu. Tato část čistícího procesu zahrnuje využití ultrafialového záření pro dezinfekci vody. V rámci pátého procesu (P5) je využito reverzní osmózy pro odstranění anorganické nečistoty. V závěrečném šestém procesu (P6) je vyčištěná voda uschována v nádržích, kde je připravena k distribuci pomocí potrubí. Jestli je kvalita vody pořád nedostačující tak je možné celý proces čištění opakovat. [44]



Obr. 11: Architektura čističky odpadních vod. [44]

Systém SWaT (Secure Water Treatment) sestává z řady senzorů, akčních členů, PLC, HMI, SCADA pracovních stanic. Celý systém využívá dvě úrovně základní komunikační sítě. První úroveň sítě vychází z hvězdicové topologie, kde ústředním prvkem je SCADA systém, který komunikuje s PLC odpovídající každému z popisovaných procesů. Druhá úroveň sítě využívá kruhovou topologii pro každé PLC. V rámci těchto částí sítě jsou situovány akční členy a senzory, které komunikují s vyhrazeným PLC. [44] Využívané atributy síťového provozu jsou prezentovány v Tab. 4. Tyto atributy nemají v tomto případě strukturu, a proto jsou uvedeny v tabulce bez závislosti na řádku a sloupci tabulky.

Tab. 4 – Využité atributy síťového provozu v rámci datasetu 2. [44]

Síťové atributy			
IP serveru	Typ logu	Typ síťového rozhraní	Směrování dat
Zdrojová proxy	Modbus - kód funkce	Modbus - popis funkce	Modbus - ID transakce
Zdrojová IP	Cílová IP	Protokol	Jméno aplikace
Modbus - ID senzoru nebo akčního členu	Modbus - přenesená hodnota	Zdrojový port	Zdrojový port

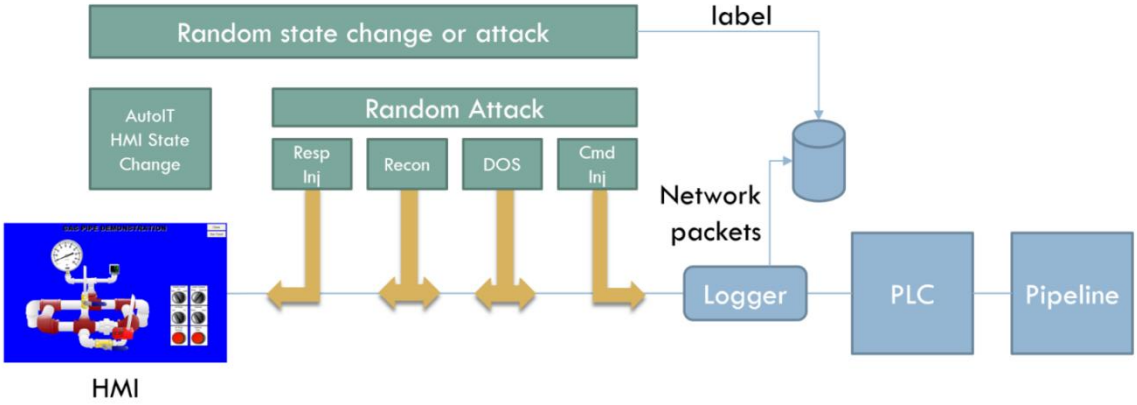
Dataset obsahuje chod čističky odpadních vod SWaT po několik dní. V průběhu záznamu bylo na popisovaný systém provedeno několik kybernetických útoků, které trvaly od několika minut do několik desítek minut.

Záznam této činnosti byl poskytnut vědecké komunitě. Tento dataset byl využit v rámci disertační práce pro tvorbu modelů strojového učení a jejich testování. [44]

Využito bylo šest kybernetických útoků pro ověření zvoleného řešení. Pro lepší orientaci bylo využito uniformní značení ve formě **CA2_1** až po **CA2_6** pro každý z kybernetických útoků. Kybernetický útok **CA2_1** ovlivňuje senzor nádrže pro znečištěnou vodu, čímž způsobuje vyčerpání vody pomoci pumpy. Tato pumpa je následně poškozena. Kybernetický útok **CA2_2** modifikuje senzor pro proplachování (změna úrovně kontrolovaného tlaku). Proces proplachování se ve výsledku cyklicky opakuje, což zapříčiňuje zastavení běžných operací systému. V rámci kybernetického útoku **CA2_3** je změněno nastavení senzoru pro kontrolu UV záření. V závislosti na tom je UV záření vypnuto a taktéž i navazující pumpa. Kybernetický útok **CA2_4** způsobuje přenastavení senzoru pro kontrolu hladiny v nádrži. To má za následek přetečení nádrže. Kybernetický útok **CA2_5** má za následek zastavení pump pro dávkování chemikálií (HCl a NaOCl). V důsledku čehož nedochází k dostatečnému čištění vody. Kybernetický útok **CA2_6** zapříčiňuje zastavení pumpy, která přečerpává znečištěnou vodu do druhé fáze (P2). V důsledku toho je zastaven průtok vody. Těchto šest kybernetických útoků bylo využito pro nastavění a testování systému detekce anomálií. Pro ověření vytvořeného systému byly zvoleny tři kybernetické útoky **CA2_7**, **CA2_8** a **CA2_9**. Tyto tři kybernetické útoky nebyly využity pro tvorbu systému pro detekci anomálií, a proto představují vhodné testovací datasety pro ověření detekčních schopností představeného systému. Kybernetický útok **CA2_7** je zaměřen na přečerpávací pumpy. Výsledkem tohoto kybernetického útoku je prasknutí vodovodního potrubí. Kybernetický útok **CA2_8** je zaměřen na tlakový senzor řídicí proces oplachu. Tento kybernetický útok zapříčinil nepřetržité opakování tohoto procesu, přičemž byl zaznamenán pokles vody v nádržích. Kybernetický útok **CA2_9** je zaměřen na zastavení pump přečerpávající vodu v rámci čističky odpadních vod. [44] Tyto kybernetické útoky byly využity v kapitole (6.2).

5.3.3 Plynovod ICS dataset

Tento systém byl vybudován na „Mississippi State University“ a svou funkčností představuje plynovod. Navíc byl poprvé publikován v článku [45]. Pro přehlednost je tento dataset v disertační práci dále pojmenován jako **dataset 3**. Popisovaný systém sestává ze tří základní částí. První částí jsou senzory a akční členy. Jako akční členy jsou využity pumpa a solenoid pro kontrolu tlaku v potrubí. Samotné plynové potrubí funguje v rámci tří módů. Jedná se automatický mód, manuální mód a vypnutí potrubí. V případě automatického módu je využíváno dvou schémat regulace plynovodu. První využívá regulace pomocí pumpy a druhé schéma využívá regulace pomocí solenoid. [45] Na Obr. 12 je znázorněna architektura využitého systému plynovodu.



Obr. 12: Architektura plynovodu. [45]

Do druhé části je zařazena samotná komunikační síť, která využívá Modbus RTU v sériovém režimu. Pakety přenášené přes komunikační síť obsahují adresu zařízení, kód funkce, „payload“, CRC (Cyclical Redundancy Check), LRC (Longitudinal Redundancy Check). V rámci poslední části systému je využito MTU jako centrálního bodu ICS. Ten komunikuje s několika RTU, kterým distribuuje příkazy. Ty naopak zasílají zprávy zpět MTU. Celý systém je možné konfigurovat skrze HMI. [45] Využité síťové a procesní atributy pro sestavení modelů strojového učení jsou zaznamenány v Tab. 5. Tyto atributy nemají v tomto případě strukturu, a proto jsou uvedeny v tabulce bez závislosti na řádku a sloupci tabulky.

Tab. 5 – Využité atributy síťového provozu v rámci datasetu 3. [45]

Síťové a procesní atributy		
Adresa „Slave“ zařízení	Kód funkce	Délka Modbus rámce
Přednastavená hodnota – žádaná hodnota (tlak)	PID - gain	PID – reset rate
PID – dead band	PID – čas cyklu	PID - rate
Využitý mód systému	Využité schéma pro kontrolu systému	Stav pumpy
Stav solenoidu	Měření tlaku v potrubí	CRC
Atribut indikující příkaz nebo odpověď		

Použito bylo šest kybernetických útoků pro ověření zvoleného řešení. Pro lepší orientaci bylo využito uniformní značení ve formě **CA3_1** až po **CA3_6** pro každý z kybernetických útoků. Kybernetický útok **CA3_1** je zaměřen na změnu přednastavených hodnot (žádaných hodnot) tlaku v rámci plynového potrubí. Tento typ útoku zapříčiňuje anomální chování systému. Druhý kybernetický útok

CA3_2 ve svém důsledku mění „reset rate“ v rámci PID (Proportional, Integral, and Derivative – kontrolér regulovaného procesu). Kybernetický útok **CA3_3** je zaměřen na změnu času cyklu v rámci PID. U kybernetického útoku **CA3_4** je náhodně měněn mód systému. Tento útok lze svou podstatou zařadit do oblasti injekce škodlivých příkazů. Výsledkem pátého kybernetického útoku **CA3_5** je posunutí systému do kritického stavu, který není pro něj běžný. Kybernetický útok **CA3_6** je založen na přeposílání velkého množství paketů se špatnými CRC hodnotami, což vede k DoS útoku. Těchto šest kybernetických útoku bylo využito pro nastavění a testování systému detekce anomálií. Pro ověření vytvořeného systému byly zvoleny tři kybernetické útoky **CA3_7**, **CA3_8** a **CA3_9**. Tyto tři kybernetické útoky nebyly využity pro tvorbu systému pro detekci anomálií, a proto představují vhodné testovací datasety pro ověření detekčních schopností představeného systému. Kybernetický útok **CA3_7** je stejně jako kybernetický útok **CA3_1** zaměřen na změnu přednastavených hodnot (žádaných hodnot) tlaku v rámci plynového potrubí. Kybernetický útok **CA3_8** je zaměřen na manipulaci hodnot tlaku při jejich snímání. Poslední kybernetický útok **CA3_9** je zaměřen na úpravu hodnot tlaku, které jsou poslány na zařízení „Master“. [45] Tyto kybernetické útoky byly využity v kapitole 6.2.

5.4 POPIS ZVOLENÝCH ALGORITMŮ

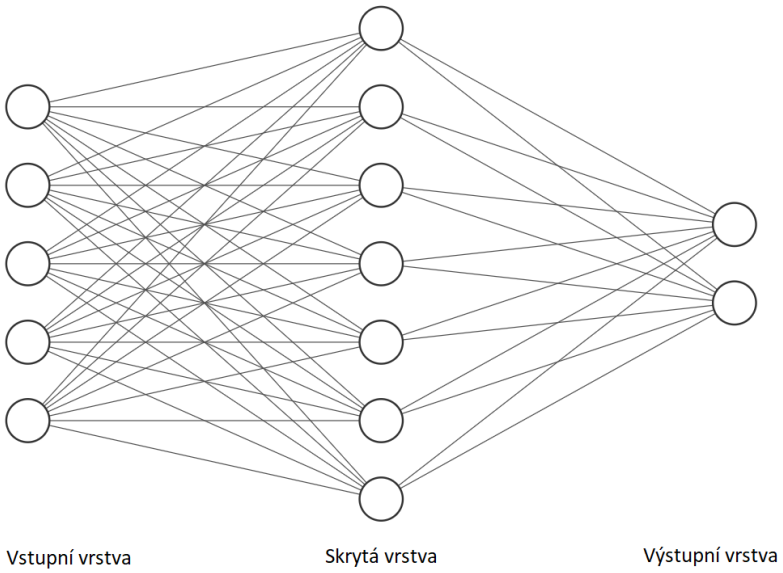
V rámci této podkapitoly byly názorně popsány všechny algoritmy strojového učení, které byly využity pro návrh řešení systému pro detekci anomálií, který je experimentálně zhodnocen v kapitole 6.

5.4.1 Základní neuronová síť

Neuronové sítě (ANN) vycházejí svojí filozofií z fungování řídicího centra nervové soustavy u biologických organismů. Samotné neurony jsou jedinečné buňky předurčené pro uchování a přenos informací nezbytných pro správný chod organismu. Základ moderní neuronové sítě položil Rosenblatt ve své publikaci [46], kde představil perceptron. Jedná se o jednoduchý příklad dopředné neuronové sítě s jedním neuronem. Omezené vlastnosti perceptronu byly odstraněny modifikací zvanou vícevrstvý perceptron, jehož architektura je využívána dodnes.

Neuronová síť je obvykle sestavena z několika vzájemně propojených neuronů v rámci jednotlivých vrstev. Neuronová síť obsahuje tři základní části, které jsou zobrazeny v Obr. 13. Vstupní vrstva má stejnou dimenzi jako počet využívaných atributů. Dá se říct, že vstupní vrstva je určena pro předání informací o řešeném problému skrytým vrstvám. V rámci vstupních vrstev neprobíhají žádné výpočty. Počet skrytých vrstev může být od jedné až po stovky. Právě na počtu a struktuře skrytých vrstev záleží, do jaké míry bude vytvořený model efektivní, tedy jak efektivně bude neuronová síť naučena reprezentaci dat. Vstupní vrstva předá signály na vstup první skryté vrstvy. V rámci této vrstvy každý z neuronů

vypočítá svůj výstup a předá jej jako vstup další skryté vrstvě. Tento proces se opakuje v závislosti na počtu skrytých vrstev. Výstupní vrstva je podobná jako vstupní, avšak oproti ní je výstupní vrstva využívána pro predikci. V případě klasifikace se jedná o predikci jedné z definovaných tříd.



Obr. 13: Základní neuronová síť. [vlastní zdroj]

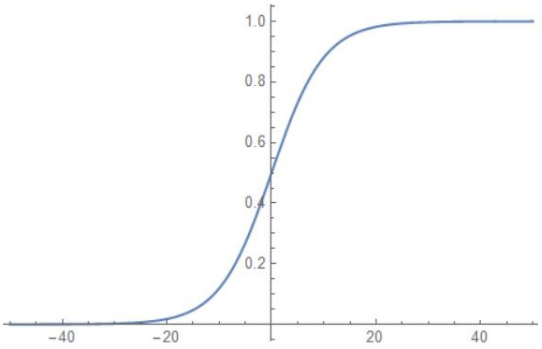
Operace v rámci jednotlivých neuronů využívají operace s maticemi, kde výpočet výstupů neuronu závisí na vztahu (6). Základem je multiplikace každého vstupu $\mathbf{x_i}$ (datového záznamu) s maticí vah $\mathbf{w_i}$ vstupů do neuronu. Váhy neuronu představují míru významnosti vstupu pro řešený problém. Práh je zastoupen proměnou **b** (bias) pro neuron a umožňuje větší variabilitu modelu tak, aby odpovídal datasetu. [46]

$$\mathrm{Y} = A(\sum_{i=1}^{n} x_i . w_i + b) \tag{6}$$

Práh spolu s váhami a vstupy tvoří vnitřní potenciál neuronu. Tento potenciál vychází z biologických neuronů, které po příjmu informace excitují a vyšlou informaci nebo v opačném případě zůstanou nečinné. Aktivační funkce (**A**) v rámci umělé neuronové sítě na základě vnitřního potenciálu neuronu rozhoduje, zda neuron bude aktivován nebo ne, a tudíž zdali vyšle informaci (výstup). [46]

$$\mathrm{A(x)} = \frac{1}{1+e^{-x}} \tag{7}$$

Hojně využívanou aktivační funkcí je sigmoidální funkce, její matematický zápis je formulován ve vztahu (7). Podle průběhu sigmoidální funkce zobrazené na Obr. 14, je zřejmé, že nabývá hodnot 0 a 1, kde hranice pro aktivaci je 0,5.



Obr. 14: Sigmoidální aktivační funkce. [vlastní zdroj]

Dosud popsané procesy neuronové sítě se vztahují k propagaci informací od vstupní vrstvy až k výstupní vrstvě. Učení neuronové sítě je však cyklický proces, který lze definovat jako optimalizační úkol. Pro registraci chyb při trénování a jejich využití pro nastavení modelu je využíváno algoritmu „Back-propagation“. Tento algoritmus v první fázi porovná skutečný výsledek ($\mathbf{Y}^{\mathbf{real.}}$) s výsledkem, který poskytuje (predikuje) model ($\mathbf{Y}^{\mathbf{predik.}}$). Jedním ze základních algoritmů, využívaných pro výpočet chyby, je střední kvadratická chyba (Mean Squared Error) neboli **MSE** (8). Základní předpoklad pro velikost MSE vyplývá z velikosti rozdílu středních hodnot sledovaných výsledků. Čím je toto číslo menší, tím více je model přesnější. Tímto výpočtem získáme tzv. chybovou funkci (Loss function), která vyjadřuje, jak dobře model odpovídá datům pro trénování. [46]

$$\mathrm{MSE} = \frac{1}{2}\sum_{i=1}^{n}\left(Y_i^{real.} - Y_i^{predik.}\right)^2 \tag{8}$$

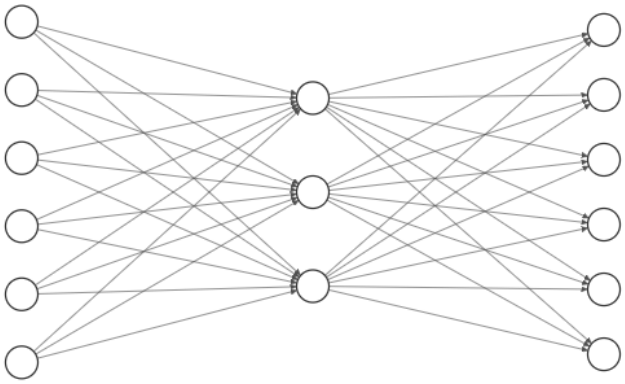
Algoritmus Back-propagation využívá MSE jako funkci, kterou musí optimalizovat. V tomto případě je hodnota MSE uvažována jako chybová funkce, kterou je nutné minimalizovat.

Z tohoto důvodu optimalizační algoritmus využívá vypočtenou chybu, kterou se snaží zpětně šířit. Od výstupní vrstvy, ke vstupní vrstvě. Přitom koriguje váhy a prahy jednotlivých neuronů. K tomu je využito parciálních derivací chybové funkce (řetízkové pravidlo) podle jednotlivých vah (9), kde **f** představuje chybovou funkci a **w** představuje váhy i-tého neuronu v j-té vrstvě. Základním předpokladem pro naplnění této funkce je její konvergence do globálního maxima. [46]

$$\nabla \mathrm{f} = \frac{\partial f}{\partial w_{i,j}} \tag{9}$$

5.4.2 Autoenkoder

Z pohledu detekce anomálií je vhodné využít architekturu autoenkodéru. V minulosti spočívalo jeho základní využití především v odstranění šumu ze vstupních signálů, díky jeho generalizačním schopnostem. Ty jsou však dobře využitelné pro vytvoření modelu normálního chování sledovaného systému s následnou detekcí anomálií. Hlavní myšlenkou autoenkodéru je vytvoření výstupů, které jsou identické se vstupy, tedy komprese a dekomprese dat. Z tohoto důvodu je dimenze vstupní vrstvy neuronové sítě stejná jako dimenze výstupní vrstvy. V rámci neuronové sítě se nachází tzv. „hrdlo“ (bottleneck), které má minimální dimenzi neuronů oproti ostatním vrstvám autoenkodéru viz. Obr. 15. Přítomnost hrdla v neuronové síti slouží ke generalizaci a naučení jen významných informací. Množina vrstev od vstupní vrstvy po hrdlo se nazývají enkodér. Oproti tomu, vrstvy od hrdla až k výstupní vrstvě nazýváme dekodér.



Obr. 15: Struktura autoenkodéru. [vlastní zdroj]

Jak neuronová síť, tak rekurentní LSTM síť využívá autoenkodér. Autoenkoder je typ neuronové sítě pro řešení regresních úloh. Z tohoto důvodu je nutné vyřešit problematiku určení hranice mezi regulérním provozem sledovaného systému a anomáliemi. Proto je vytvořen specifický postup pro definici této hranice.

Definování finální hranice se skládá z několika kroků. Nejprve je každá predikovaná hodnota srovnána s reálnými daty obsahující kybernetické útoky. Rozdíl těchto hodnot nám určuje, do jaké míry skutečná data vybočují od modelu, což je poté možno klasifikovat jako anomálii. Uvažujeme-li tedy dvě hodnoty, první připadá modelu $\boldsymbol{x_{i,j}^{mod}}$ a druhá hodnota definovaná jako $\boldsymbol{x_{i,j}^{real}}$ přináleží testovaným datům. Poté výsledný vztah lze definovat jako:

$$x^r = \left| x_{i,j}^{mod} - x_{i,j}^{real} \right| \tag{10}$$

kde hodnota x přináleží i-tému příznaku v j-tém datovém bodu. Pro výpočet výsledné hodnoty odchylky pro každý z datových bodů je využito MSE vzorce viz vztah (11).

$$\text{MSE} = \frac{1}{2}\sum_{i=1}^{n}(x_i^r)^2 \quad (11)$$

Výsledek reprezentuje průměr odchylek dílčích příznaků v rámci jednoho datového bodu. Získané hodnoty pro jednotlivé datové body reprezentují míru vzdálenosti od vytvořeného modelu. Je však nutné definovat jasnou hranici, která rozděluje tyto hodnoty do oblastí normálního nebo abnormálního síťového provozu.

Často využívanou metodou k určení hranice je ROC (Receiver Operating Characteristic). Ta však vykazuje nevýhody pro nevybalancovaný dataset. V tomto případě nelze tuto metodu využít, jelikož oblast detekce anomálií vychází z předpokladu, který definuje anomálii jako událost, jejíž výskyt je vzácný. Proto se ve využitém datasetu nachází několikanásobně více záznamů pro normální síťový provoz než kybernetických útoků. Tudíž dataset není vybalancován. Z tohoto důvodu je vhodné využít Precision/Recall křivku pro definici bodu, který je hranicí nutnou pro rozlišení mezi kybernetickým útokem a normálním provozem.

Tato metoda vybírá hraniční hodnotu ze zvolené množiny hodnot, přičemž ke každé vybrané hodnotě vypočítává dva parametry: Precision, také nazývaný jako pozitivně predikovaná hodnota (Positive Predictive Value – **PPV**) a Recall, taktéž nazýván skutečná pozitivní hodnota (True Positive Rate – **TPR**).

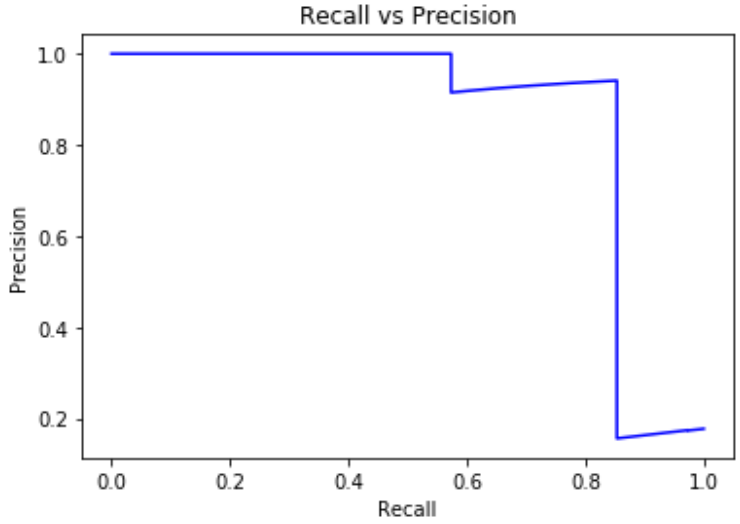
Tyto dva parametry charakterizují detekční schopnosti každé ze zvolených hranic. Parametr TPR vyjadřuje, jak dobře zvolená hranice umožňuje detekovat pozitivní třídu (normální síťový provoz), vzhledem k chybě v klasifikaci (klasifikace kybernetického útoku jako normální provoz). Oproti tomu PPV vyjadřuje, jak dobře zvolená hranice umožňuje detekovat pozitivní třídu, vzhledem k druhé chybě v klasifikaci, tedy klasifikace normálního provozu jako kybernetický útok. Oba tyto parametry je vhodné maximalizovat pro zajištění ideálních vlastností zvoleného řešení.

Maximalizace obou ze zmíněných parametrů však není tak snadná, jak by se na první pohled zdálo. Pří maximalizaci jednoho parametru většinou od určité hodnoty druhý klesá a naopak. Proto je nutné definovat graf, ve kterém je zobrazena Precision/Recall křivka pro různé hodnoty hranic, aby bylo možné vybrat optimální řešení.

Z Obr. 16 je patrný průběh této křivky, který obecně kopíruje horní a pravou stranu grafu. Optimální hodnota hranice se poté nachází v horním pravém rohu popisovaného grafu. Závěrečným krokem je definice postupu, který automaticky povede k výběru právě jedné optimální hodnoty pro definici referenční hranice pro klasifikaci budoucích dat. Základním předpokladem pro nalezení optimální hodnoty je maximalizace obou parametrů PPV a TPR. Pro určení optimálního řešení byl definován předpoklad o nejnižším rozdílu mezi využitými parametry.

Platí tedy, že při snižujícím se rozdílu (minimalizaci) mezi parametry PPV a TPR se zvyšují schopnosti navrhovaného řešení definovat optimální hranici (H^{opt}) pro rozdělení normálního provozu a kybernetických útoků. Máme-li množinu měření hranic $h = \{i \in \mathbb{Z} | i \neq 0\}$, tak kalkulujeme rozdíl pro každý případ z množiny podle vztahu (12). Poté hledáme minimum mezi získanými výsledky v získané množině.

$$H_i = |TPR_i - PPV_i| \tag{12}$$



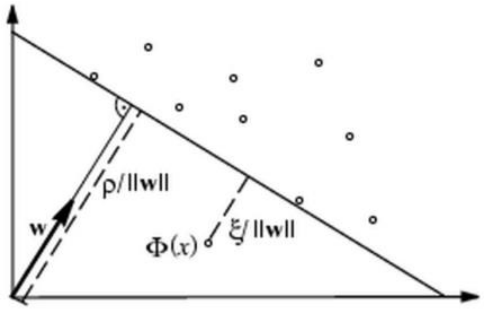
Obr. 16: Precision/Recall křivka pro různé hranice. [vlastní zdroj]

5.4.3 OCSVM

Jedná se o klasifikační algoritmus kombinující učení s učitelem a učení bez učitele. One-class Support Vector Machines (OCSVM) je modifikací algoritmu podpůrných vektorů neboli Support Vector Machine (SVM). Základem klasického SVM je vytvoření nadroviny, která odděluje data. Tuto nadrovinu je nutné maximalizovat, tedy oddělit od sebe dvě skupiny dat. Základní rovnice, vyjadřující tuto nadrovinu pro lineárně separovatelná data je zobrazena ve vztahu (13). [47]

$$f(\bar{x}) - \xi = \bar{w}\bar{x} + b \tag{13}$$

Kde $\bar{\boldsymbol{w}}$ je normála nadroviny, $\bar{\boldsymbol{x}}$ vyjadřuje vektor hodnot, b určuje umístění podpůrných vektorů a $\boldsymbol{\xi}$ je tzv. přídavná proměnná "slack variables", která určuje hodnotu pro každý datový bod, který se nachází mimo vytyčené území. Specifickou modifikací SVM algoritmu je OCSVM. Tento algoritmus se nesnaží oddělit data mezi sebou, ale je určen pro vytvoření nadroviny mezi počátkem souřadné soustavy a daty, kterou se snaží maximalizovat z důvodu vytvoření klasifikačního modelu založeného pouze na jedné třídě (normálním chování systémů) viz. Obr. 17. [47]



Obr. 17: Reprezentace OCSVM. [47]

Nicméně bylo nutné pracovat především s daty, která nelze lineárně separovat. Z tohoto důvodu bylo nutné převést data do vyššího dimensionálního prostoru pomocí tzv. „jádrové transformace“ (kernel trick) podle vztahu (14 a 15). [47]

$$\Phi: R^d \rightarrow \mathcal{H} \tag{14}$$

$$K(x_i, x_j) = (\emptyset(x_i).\emptyset(x_j)) \tag{15}$$

Vztah reprezentující OCSVM je znázorněn v rovnici (16), kde **k(xᵢ,x)** reprezentuje jádrovou transformaci. Pro splnění tohoto úkolu byla zvolena speciální radiální jádrová transformace, kvůli jejím vhodným vlastnostem. Její plné matematické znění je znázorněno v rovnici (17). [47]

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x) - \rho \tag{16}$$

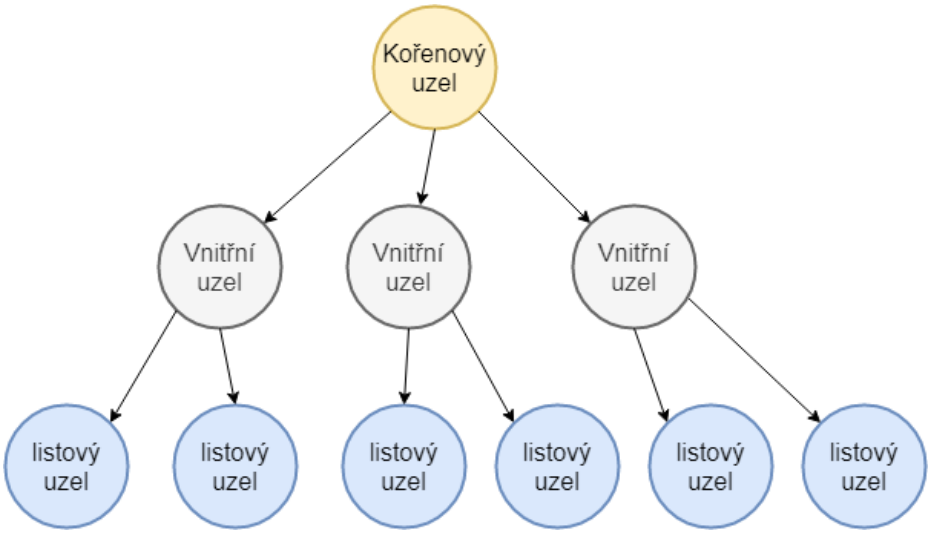
$$K(x_i, x) = exp(-\gamma \|x_i - x\|^2), \;\; \gamma > 0 \tag{17}$$

Kde $\mathbf{x_i}$ reprezentuje datové body, **x** je orientační bod a $\boldsymbol{\gamma}$ zde reprezentuje gamma parametr pro optimální určení velikosti nadroviny. Tento parametr je využívaný u nelineární klasifikace. Určuje, jak daleko dosahuje vliv každého trénovacího příkladu. Jinými slovy gamma parametr definuje, jestli budou mít i vzdálené případy vliv na konstrukci hranic nadroviny. Z tohoto důvodu je nutné správně nastavit tento parametr, aby došlo k vytvoření optimálního prediktivního modelu. **Parametr gamma** je využit v rámci optimalizace OCSVM algoritmu. [47]

5.4.4 Isolation Forest

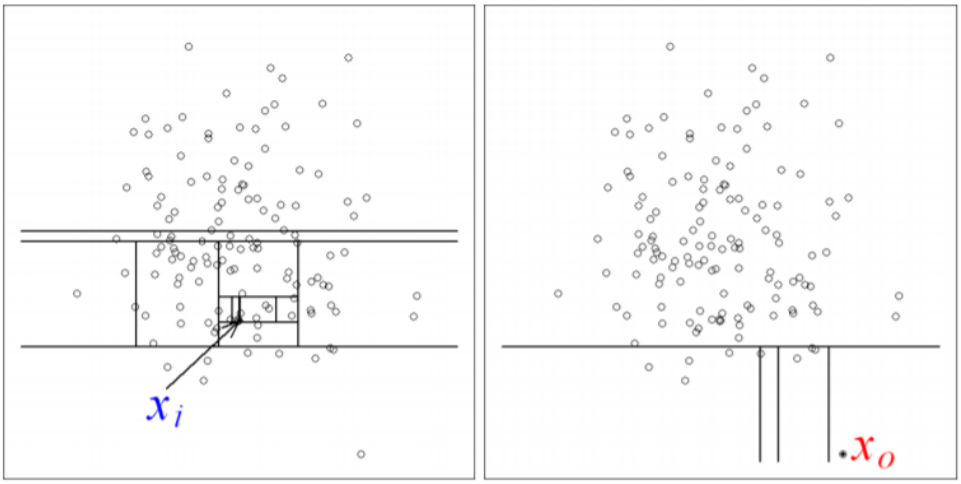
Tento algoritmus strojového učení fundamentálně vychází z algoritmu „Random Forest“ (RF). Tento algoritmus byl představen autorem Liu v publikaci [48] Tento autor se v případě algoritmu „Isolation Forest“ (IF) vydal odlišným. směrem než většina dosavadních řešení v oblasti detekce anomálií. Tato

standartní řešení vycházejí z definování modelu reprezentujícího normální procesy sledovaného systému. Tento model je poté využit k detekci anomálií. IF vychází ze dvou předpokladů. Prvním z předpokladů vychází ze skutečnosti, že anomálie jsou zastoupeny v datech velmi zřídka. Druhý předpoklad vychází z rozdílnosti hodnoty mezi atributem normálního záznamu a atributem anomálního záznamu. Samotný algoritmus je založen na stromové struktuře, přesněji na algoritmu rozhodovací strom (decision tree – DT). Tento algoritmus vytváří strukturu podobnou diagramu, který je zobrazen v Obr. 18. V rámci kořenového uzlu jsou zastoupena všechna data využitá pro trénování modelu. Na základě pravidel jsou tato data rozdělena do vnitřních uzlů, které reflektují rozdílné skupiny dat. Tyto skupiny mohou být dále děleny pomocí dalších vnitřních uzlů. Jestliže algoritmus dělením vnitřních uzlů nezíská další informační užitek, tak je skupina dat uložena v listových uzlech. Ty představují konečné skupiny dat. Základní předpoklad pro IF je založen na separaci anomálií. Ty jsou situovány blíže kořenového uzlu, zatímco data vztahující se k normálnímu provozu jsou uložena hlouběji ve stromové struktuře. IF využívá řady stromů, ze kterých vytváří les jako prediktivní model. [48]



Obr. 18: Stromová struktura. [vlastní zdroj]

IF oproti algoritmům pro profilování normálního provozu je zaměřen na kvantitativních vlastnostech anomálií, podle kterých jsou poté identifikovány. Tedy IF přímo identifikuje anomálie a nezabývá se profilováním normálního chování. Jelikož anomálie nabývají unikátních hodnot, které vybočují a jsou často ojedinělými body v rámci datasetu, tak se předpokládá jejich separace do listových uzlů ve velmi krátkém čase. [48] Ukázka separace normálních bodů je ilustrována v Obr. 19, kde x_i představuje záznam normálního provozu a bod x_0 představuje zaznamenanou anomálii.



Obr. 19: Izolace datových bodů pomocí IF. [48]

Z obr. 18 je patrný vztah mezi unikátními (extrémními) hodnotami, které se dají charakterizovat jako anomálie a vzdáleností listového uzlu od kořenového uzlu, do kterého budou tyto body zařazeny. Identifikace anomálií je tudíž závislá na vzdálenosti listového uzlu od kořenového uzlu.

V následujících odstavcích jsou popsány hyperparametry pro optimalizaci algoritmu IF. Jejich nastavení má značný vliv na výsledky a výkon vybraného algoritmu strojového učení. Tyto hyperparametry jsou využity pro optimalizaci zvoleného řešení. Mezi základní hyperparametry patří:

1. **Maximální počet atributů** – jedná se o hyperparametr, který definuje maximální počet atributů využitých pro každé dělení v rámci stromu. Pro každé dělení jsou vybrány náhodné atributy respektující definovaný počet v rámci tohoto hyperparametru. Tento hyperparametr je využíván ke kontrole a omezení "přeučení" modelu.

2. **Počet vzorků** – tento hyperparametr definuje maximální počet vzorků(záznamů), který je využit pro vytvoření jednoho stromu v rámci algoritmu IF.

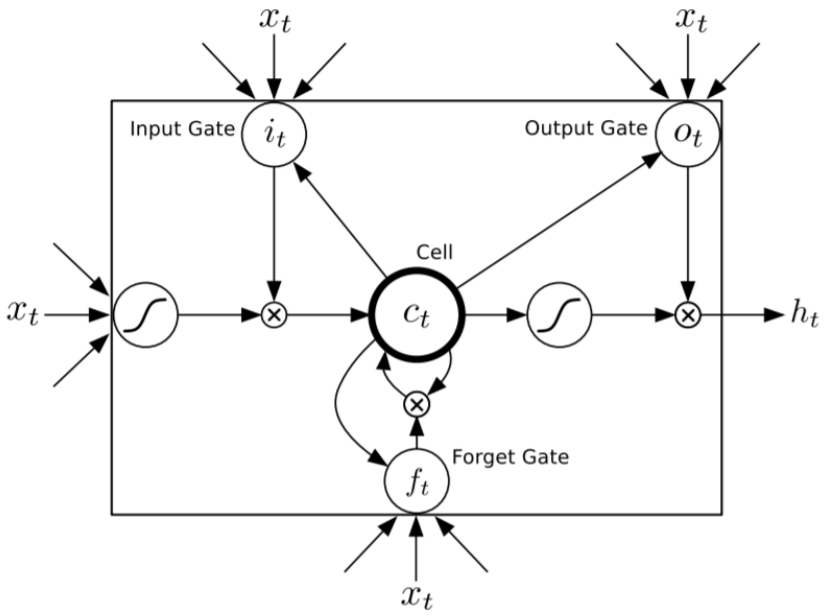
3. **Počet stromů** – hyperparametr, který definuje počet stromů, ze kterých je tvořen IF. V případě IF mluvíme o tzv. lesu, který se skládá z jednotlivých stromů.

4. **Kontaminace** – tento parametr vyjadřuje odhad kontaminace datasetu prostřednictvím anomálií pro danou oblast nasazení algoritmu IF. Pomocí tohoto parametru je nastavena citlivost IF pro detekci anomálií.

5.4.5 LSTM

LSTM (Long Short-Term Memory) je algoritmus strojového učení, který spadá do podskupiny rekurentních neuronových sítí. Tyto algoritmy pracují se sekvenčními daty, kde významnou roli hrají nejenom hodnoty jednotlivých atributů, ale také jejich uspořádání. LSTM je známý algoritmus, který se často využívá pro potřeby klasifikace textu nebo zvuku. Na rozdíl od klasické neuronové sítě obsahuje LSTM pozměněnou neuronovou buňku, která obsahuje tři tzv. brány, které umožňují uchovávat a přenášet informace z předešlých záznamů. [49]

V rámci této disertační práce je využita LSTM neuronová síť, kterou představili autoři Hochreiter a Schmidhuber ve své publikaci.[50] Autoři definovali silné stránky popisovaného algoritmu strojového učení. Zvláště vyzdvihli schopnost LSTM zachování znalosti v čase po relativně dlouhou dobu, při vyřešení problematiky „Vanishing Gradient Problem“. Ten nastává u vícevrstvých neuronových sítí. LSTM využívá svých předností především v rámci datasetů, které obsahují časové řady.



Obr. 20: Buňka LSTM. [51]

Jak je možno vidět v Obr. 20, buňka LSTM obsahuje předem definovanou strukturu, která je reprezentována prostřednictvím tzv. „brán“ (gates). Každá z těchto brán vykonává rozdílné funkce, které jsou nutné pro úspěšnou aplikaci LSTM algoritmu. Vstup z datasetu je znázorněn jako $\mathbf{x_t}$, který spolu se skrytým vektorem $\mathbf{h_{t-1}}$ (z předešlé buňky) vstupuje do buňky LSTM. Vstupní brána $\mathbf{i_t}$ filtruje významné vstupy od nevýznamných a kontroluje stav buňky. Výstupní brána $\mathbf{o_t}$ kontroluje, jaký výstup bude vyslán z LSTM buňky a zároveň kontroluje skrytý vektor $\mathbf{h_t}$. Zapomínací brána $\mathbf{f_t}$ definuje, která data se mají využít v rámci LSTM buňky. Navíc také definuje stav samotné LSTM buňky.[51]

V následujících odstavcích jsou popsány hyperparametry pro optimalizaci algoritmu LSTM a základní neuronové sítě. Jejich nastavení má značný vliv na výsledky a výkon zvolených algoritmů. Tyto hyperparametry jsou využity pro optimalizaci zvoleného řešení.

1. **Počet neuronů** – tento hyperparametr definuje počet neuronů v rámci jedné skryté vrstvy neuronové sítě

2. **Počet vrstev** – jedná se o hyperparametr určující počet vrstev v rámci neuronové sítě.

3. **Počet neuronů pro "bottleneck"** – z důvodu využití architektury autoenkodéru bylo nutné v rámci neuronových sítí definovat počet neuronů, kterých bude bottleneck nabývat. V rámci zvoleného řešení byly využity dva typy vrstev pro bottleneck. První typ vrstev představuje nejužší bod v celé síti (nejméně neuronů). Druhý typ vrstev představuje dvě vrstvy, které mají vygenerovány počet neuronů v rozmezí mezi vrstvou prvního typu a vrstvami pro zbylé vrstvy neuronové sítě. Oba typy vrstev pro bottlenek jsou náhodně generovány.

4. **Počet epoch** – tento hyperparametr definuje počet epoch, kterým musí každá neuronová síť projít v rámci jejího učení. V rámci jedné epochy neuronová síť prochází cyklickým procesem, kde se učí algoritmus na trénovacím datasetu a je hodnocen pomocí validačního datasetu. Tento postup zajišťuje vyšší generalizaci výsledného modelu. V rámci každé epochy je trénovací dataset rozdělen do dávek, které jsou poté vkládány do neuronové sítě.

5. **Velikost dávky** (batch) - velikost dávky definuje počet záznamů vložených do neuronové sítě předtím, než budou upraveny parametry neuronové sítě a proveden výpočet chyby sítě.

6. **Velikost "dropout"** – jedná se o hyperparametr, který je implementován v rámci neuronových sítí. Zamezuje tzv. "přeučení" neuronové sítě. Tedy neuronové sítě budou efektivně operovat v rámci trénovacích dat a budou mít problémy s novými vzory. Z tohoto důvodu je využíván hyperparametr dropout, který s určitou pravděpodobností vybírá, jaké neurony v neuronové síti budou aktivovány. Tato technika prakticky zapříčiňuje celý proces učení více náhodným.

7. **Velikost rekurentního "dropout"** – tento hyperparametr je jediným hyperparametrem, který není využit v případě základní neuronové sítě. Rekurentní dropout je fundamentálně stejný jako dropout definovaný v předchozím odstavci. Rozdílem je jen jeho aplikace na rekurentní propojení v tomto případě v rámci LSTM algoritmu.

8. **Aktivační funkce** – aktivační funkce definuje za jakých podmínek je každý z neuronů v neuronové síti aktivován a tedy definuje, jestli bude informace přicházející jako vstup do neuronu přenesena. Bez aktivační funkce by neuronová síť degradovala na lineární funkci, čímž by se také limitovala možnost řešit komplexní úlohy.

9. **Optimalizační algoritmus** – optimalizační algoritmy jsou v rámci neuronových sítí využívány k výpočtu a zpětné propagaci výsledné chyby učení (rozdíl mezi skutečností a výsledky neuronové sítě) pro upravení hodnoty vah neuronové sítě. Tento hyperparametr tedy ovlivňuje proces učení každé neuronové sítě.

10. **Míra učení** – jedná se o hyperparametr, který ovlivňuje jak rychle se dotyčný algoritmus "učí". Při vysokých hodnotách tohoto hyperparametru jsou měněny váhy dotyčných algoritmů rychleji, a proto se učí rychleji. Avšak se zvyšujícími se hodnotami se také zvyšuje riziko uváznutí v lokálním minimu.

5.5 OPTIMALIZACE

Optimalizace je proces, ve kterém je využito řady technik a postupů pro vyhledání nejlepšího, tedy optimálního řešení. Při reálném využití se snaží optimalizační algoritmy nacházet minimální nebo maximální řešení podle zadání (jedná-li se o maximalizační, nebo minimalizační úlohu). Optimalizaci lze v tomto případě definovat jako iterativní proces se zpětnou vazbou, při kterém je hledáno nejvýhodnější nastavení vybraného systému. Tedy, takové nastavení, při kterém systém vykazuje nejlepší výsledky. Kochenderfer a Wheeler v jejich knize [52] popsali základní principy optimalizace. Samotný proces optimalizace přirovnali k cyklickému procesu, při kterém je v každém cyklu proveden návrh systému, který je následně ohodnocen. Na základě tohoto ohodnocení je dotyčný návrh změněn. V případě, kdy návrh dosáhl maximálního hodnocení je ponechán a označen za finální návrh. Strategie pro prohledávání vytyčeného prostoru se v rámci jednotlivých optimalizačních technik liší. Základní pojetí optimalizace pro jednu dimenzi je definován jako vztah 18, popřípadě 19 podle publikace. [52]

$$\min_{x} f(x) \quad vyhovující\ x \in \mathrm{X} \tag{18}$$

$$\max_{x} f(x) \quad vyhovující\ x \in \mathrm{X} \tag{19}$$

Kde x představuje proměnou (např. hyperparametr) pro návrh spadající do množiny X. Každá proměnná x je využita jako vstup do funkce f, která se nazývá

cílová funkce „objective function“ (OF). Tato proměnná maximalizuje nebo minimalizuje cílovou funkci podle řešeného problému.

V rámci definované problematiky jsou optimalizační techniky využity pro hledání optimální konfigurace hyperparametrů algoritmů strojového učení. Tato kapitola disertační práce je zaměřena na objasnění teoretického rámce využitých optimalizačních technik využitých v rámci provedených experimentů. Jedná se optimalizační algoritmy: Random search, Genetický algoritmus, Tree-structured Parzen Estimator.

5.5.1 Random search

Tato optimalizační metoda vychází z metody „Grid search“ (GS). GS využívá kombinatoriky, kde sestavuje množinu všech možných kombinací využívaných hyperparametrů. Výhodou této metody je vyjádření všech možných kombinací hyperparametru. Tento postup je však výpočetně, a tudíž i časově velmi náročný. Z tohoto důvodu je využívána metoda Random search (RS). Tato metoda využívá všech kombinací hyperparametrů stejně jako GS, avšak všechny je neprověřuje. Místo toho náhodně vybírá jednotlivé zástupce. Předpokládá se, že vhodné řešení z vytyčeného prostoru nalezne RS dříve než metoda GS. RS stejně jako GS nevyužívá gradientu funkce. Z tohoto důvodu je nutné nastavení, kdy má tato metoda ukončit své hledání (např. počet iterací atd.)

Tab. 6 Pseudokód využitý pro metodu RS. [vlastní zdroj]

```
Vstupy: počet_iterací, nejlepší_výsledek, množina_jedinců
nejlepší_výsledek = 0;
jedinec =0;
For iterace in počet_iterací:
    SELECT jedinec in množina_jedinců;
        IF jedinec>nejlepší;
        THEN nejlepší=jedinec;
Nejlepší_řešení=nejlepší;
```

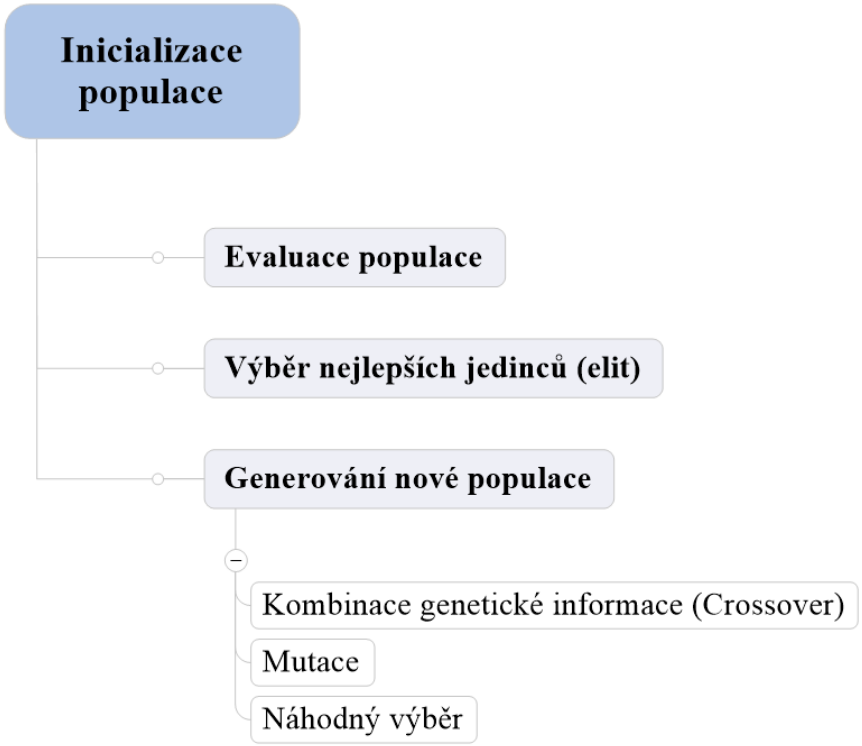
5.5.2 Genetický algoritmus

Genetické algoritmy (GA) jsou robustní vyhledávací algoritmy založené na heuristice, které vycházejí z Darwinovy evoluční teorie. Tyto algoritmy představil Holland ve své publikaci: "Adaptation in Natural and Artificial Systems" [53]. Základní myšlenka GA je založena na předpokladu, že přežijí jenom nejschopnější jedinci, kterým je umožněno se rozmnožit. Proto budou moci jejich atributy (vlastnosti) přejít na další generace. GA proto konvergují do jednoho optimálního řešení (jednotlivce), který je vybrán simulovanou evolucí. Každý jedinec je ohodnocen pomocí tzv. „Fitness Function“ (FF), také známé jako hodnotící funkce, která by se dala přirovnat k OF v rámci optimalizačních

funkcí. V případě systému pro detekci anomálií lze využít přesnost identifikace kybernetických útoku jako FF pro optimalizaci prediktivního modelu. Pro nalezení optimálního řešení je GA uzpůsoben pro minimalizaci FF. Každý jedinec je definován pomoci množiny atributů podle vzorce (20).

$$Jedinec = \{p_1, p_2, \ldots\ldots p_n\} \tag{20}$$

Tyto atributy definují jedince a také se přenášejí do následujících generací. V případě problematiky detekce anomálií pomocí algoritmů strojového učení jsou atributy definovány jako hyperparametry prediktivních modelů. Např. počet skrytých vrstev v neuronové síti, počet neuronů v jedné vrstvě nebo počet epoch pro naučení neuronové sítě atd. Základní diagram, představující fáze průběhu optimalizace pomocí genetického algoritmu, je znázorněn v Obr. 21.



Obr. 21: Hlavní části genetického algoritmu. [53]

Jak je možné vidět z Obr. 21 každý GA začíná s inicializací populace první generace. Důležité je dbát na dostatečnou diversitu atributu u jednotlivých jedinců populace. Při špatném nastavení první generace hrozí konvergování jedinců do lokálního minima namísto do globálního minima.

Druhý krok GA je zaměřen na ohodnocení každého jednotlivce a získání FF. Ve třetím kroku GA je populace rozdělena podle FF. Nejlepší skupina je klasifikována jako „elita“ a druhá méně vhodná skupina je nazvána „poraženými“. Jedinci ze skupiny elita jsou náhodně vybráni k reprodukci

nových potomků. Důležité je zajistit, aby jeden jedinec nebyl vybrán jako rodič 1 a 2. Vznik jedince je založen na tzv. Crossover, jedná se o rekombinaci dvou genetických informací za cílem vytvoření nové genetické informace. Výsledkem reprodukce je vytvoření dvou potomků, kteří náhodně zdědili atributy po svých rodičích. Poměrně důležité je využití tzv. mutace, která je využita pro náhodnou změnu jednoho z atributů z důvodu předcházení konvergence do lokálního minima. Další technikou pro předcházení lokálních minim nebo maxim je náhodná volba jedince, který je zařazen do poražené skupiny, jako jednoho z rodičů. Tyto techniky mutace a náhodné volby jsou ve vztahu k ostatním postupům využívány velmi zřídka, v rozmezí jednotek procent. [53]

Tab. 7 Pseudokód využitý pro genetický algoritmus. [vlastní zdroj]

Inputs: **hyperparametry, zachování, population$_{size}$, generace$_{velikost}$, výběr$_{náhodný}$, mutace**
Generace = 0;
Inicializace první populace (populace$_{velikost}$) ;
Jedinec = náhodný výběr parametrů;
return **(populace)**

For **generace** in **generace$_{velikost}$:**
For **jedinec** in **populace:**
Ohodnocení jedince (hodnotící funkce);
return **(hodnotící funkce)**
rozdělení seřazených jedinců (hodnotící funkce) populace vytvořená na základě parametru zachování (ponechání elit);
ponechání jedinců, kteří nebyli vybráni do skupiny elit(výběr$_{náhodný}$);

While **elity<populace:**
For **jedinec** in **populace$_{poražení}$:**
Tvorba jedinců/potomků (jedinec);
Jsou vybráni dva rodiči ze skupiny populace$_{elity}$;
Vytvoření dvou potomků prostřednictvím křížení parametrů rodičů;
Mutace parametrů potomků (mutace);
return **(potomci)**
update **(populace)**

5.5.3 Tree-structured Parzen Estimator

Tree-structured Parzen Estimator (TPE) je optimalizační algoritmus založený na Bayesovské optimalizaci, která využívá Gaussova procesu. TPE popsal autor Bergstra, ve své publikaci [54] Algorithms for hyper-parameter optimization. Tento algoritmus je poměrně častou volbou pro optimalizaci hyperarametrů pro

algoritmy strojového učení. Jeho použití je popsáno v řadě publikací, např. [55], [56], [57].

Tento optimalizační algoritmus je založen na sekvenčních modelech - Sequential model-based optimization (SMBO). Je vytvářen model, u něhož je využito sekvenčního postupu pro jeho postupné upravování. V rámci každé iterace jsou aktualizovány pravděpodobnostní hodnoty tohoto modelu. To vede k přesnějším výsledkům a postupnějšímu nalezení optimálního nastavení pro hyperparametry podle OF. Z důvodu optimalizace je vytvořen tzv. "náhradní model", který je vytvořen pomocí počáteční množiny modelů s rozlišným nastavením hyperparametrů. Výsledný náhradní model je poté subjektem optimalizace pomocí nových dat z modelů podle P(x|y). V rámci každé iterace se algoritmus rozhodne, jaké další hyperparametry vybere na základě předchozích modelů. Při startu TPE je vybrána poměrně malá množina iterací, v rámci, kterých jsou vytvořeny modely s náhodným výběrem hyperparametrů. V dalším kroku je počáteční množina rozdělena do dvou skupin podle vztahu (21). [54]

$$P(x|y) = \begin{cases} l(x) \rightarrow if \ldots y \prec y^* \\ g(x) \rightarrow if \ldots y \geq y^* \end{cases} \qquad (21)$$

Zde $\mathbf{y^*}$ vyjadřuje prahovou hodnotu (OF). Podle této hodnoty jsou jednotlivé výsledky rozděleny do dvou skupin. Skupina distribuce pravděpodobnosti l(x) představuje zástupce s nejlepším skóre. Druhá skupina distribuce pravděpodobnosti **g(x)** představuje všechny ostatní zástupce. TPE využívá na základě reálných dat věrohodností funkce pro výběr nejlepších kandidátů z první skupiny **l(x)**. TPE je poté zaměřen na nalezení optimálního řešení tedy jeho maximalizace. To lze vyjádřit jako poměr l(x)/g(x).[54]

Tab. 8 Pseudokód využitý pro TPE. [54]

Inputs: **hyperparametry**

Výběr hyperparametrů;
Vytvoření klasifikačních modelů (poměrně malá množina);
Vytvoření náhradního modelu;

For **iterace** in **počet_iterací:**
    Získání hodnoty cílové funkce podle pravděpodobnostní reprezentace;
    Mapování hyperparametrů pomocí pravděpodobností cílové funkce;
    Změna pravděpodobnostní reprezentace náhradního modelu;
return **(optimální řešení)**

5.6 MULTIKRITERIÁLNÍ HODNOCENÍ

V rámci systému pro detekci kybernetických útoků pro systémy ICS je využito technik multikriteriálního hodnocení (MH) známých také pod názvem multikriteriální analýza (MCA). Multikriteriální hodnocení je jednou z částí oblasti optimalizace. Základní myšlenkou a úkolem MH je výběr jedné varianty z množiny možností na základě vybraných kritérií. MH je využíváno zvláště při velmi obtížných rozhodovacích úlohách, ve kterých je nutno vybrat nejlepší variantu. To se děje prostřednictvím kritérií, která jsou často fundamentálně zaměřena proti sobě. Například mějme dvě kritéria, která chceme obě maximalizovat. Mezi těmito kritérii je však definován vztah. Při maximalizaci jednoho kritéria nastává minimalizace druhého a naopak. Z tohoto pohledu je nutné vybrat vhodnou variantu, která nebude nejlepší pro obě kritéria, ale zato bude určitým kompromisem. V rámci disertační práce je využito multikriteriální hodnocení pro definici OF, která je nezbytná pro optimalizační algoritmy.

V rámci řešené problematiky byla využita metoda multikriteriálního hodnocení TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution). Tato metoda byla poprvé publikována autory Tzeng, a Huang v publikaci [58]. Mezi klady této metody patří nízká výpočetní náročnost a konzistentnost metody. Tato metoda bere také v potaz nejenom nejlepší možné výsledky, ale také ty nejhorší, což umožnuje negovat špatné výsledky v jednom kritériu za dobré výsledky v jiném. V rámci metody TOPSIS je zavedena tzv. „bazální hodnota“ a „ideální hodnota“, kde bazální hodnota představuje nejhorší možné řešení v rámci zvoleného kritéria, ideální hodnota představuje nejlepší možné řešení v rámci zvoleného kritéria. Účelem této metody je maximalizovat vzdálenost od bazální hodnoty, přičemž minimalizovat vzdálenost od ideální hodnoty pro každé kritérium. [58] Postup výpočtu TOPSIS lze shrnout do následujících bodů, které popisuje ve své publikaci autor Vavrek [59]. Postup zahrnuje následující kroky:

- Vytvoření matice všech možných řešení $\mathbf{M_{m\times n}}$, kde je každý atribut v rámci jednoho řešení reprezentován jako $\mathbf{x_{i,j}}$ pro i,$x, j \in \mathbb{N}$. Kde tato matice může být znázorněna podle předpisu (22). [59]

$$M = \begin{pmatrix} x_{1,1} & \cdots & x_{1,j} \\ \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} \end{pmatrix} \tag{22}$$

- Ve druhém kroku je vytvořena normalizovaná matice $\mathbf{M_{norm}}$, ve které jsou zastoupeny jednotlivé normalizované atributy $\mathbf{N_{i,j}}$. Tato transformace je provedena podle vztahu (23). [59]

$$N_{i,j} = \frac{x_{i,j}}{\sqrt{\sum_{j=1}^{n} x_{i,j}^2}} \tag{23}$$

- Ve třetím kroku je normalizovaná matice $\mathbf{M_{norm}}$ vynásobena váhami $\mathbf{W_{i,j}}$, kde $\sum_{j=1}^{j} W_{i,j} = 1$. Tyto váhy vyjadřují významnost jednotlivých kritérií. Vzniká tzv. váhovaná normalizovaná matice podle vztahu (24). [59]

$$V_{i,j} = W_{i,j} \,.\, N_{i,j} \tag{24}$$

- Ve čtvrtém kroku je vybrána bazální hodnota $\mathbf{B_j}$ a ideální hodnota $\mathbf{I_j}$ pro každé kritérium. V případě kritéria vybraného pro maximalizaci postupujeme podle (25). [59]

$$B_j = min\, V_j \mid I_j = max\, V_j \tag{25}$$

- V pátém bodě jsou v rámci celé matice $\mathbf{V_{i,j}}$ získány hodnoty euklidovské vzdálenosti mezi bazálními hodnotami $\mathbf{B_j}$ a všemi kritérii $\mathbf{V_{i,j}}$. Ten samý postup je uplatněn i pro ideální hodnoty $\mathbf{I_j}$. Výsledné součty jsou realizovány podle vztahu (26). [59]

$$B_j^* = \sqrt{\sum_{j=1}^{n}(v_{i,j} - B_j)^2} \mid I_j^* = \sqrt{\sum_{j=1}^{n}(v_{i,j} - I_j)^2} \tag{26}$$

- Závěrečným bodem metody TOPSIS je výpočet koeficientu významnosti K pro jednotlivé varianty. Tento koeficient je vypočten pomocí $\boldsymbol{B_j^*}$ a $\boldsymbol{I_j^*}$ podle vztahu (27). [59]

$$K = \frac{B_j^*}{B_j^* + I_j^*} \tag{27}$$

5.7 HODNOCENÍ VÝSLEDKŮ

Tato kapitola popisuje metodiku hodnocení klasifikačních modelů, která je využívána v rámci odborné komunity. Podrobný popis jednotlivých metrik je důležitý pro objasnění a ohodnocení jednotlivých řešení v rámci dané problematiky. Každý z využitých algoritmů strojového učení je hodnocen pomocí vybraných metrik, přičemž souhrnné výsledky jsou publikovány v kapitole 6 této disertační práce.

Každý model strojového učení je hodnocen pomocí konfúzní matice. Ta vyjadřuje vztah mezi predikovanými třídami a skutečnými třídami. Definované třídy vyjadřují příslušnost dílčích datových bodů ve zkoumané problematice. V rámci binární klasifikace má konfúzní matice dvě třídy. První vyjadřuje normální chod systému **(negativní třída)**, druhá zase kybernetický útok na systém (**pozitivní třída**). Výsledná matice na základě testovacího datasetu, vyjadřuje, jak přesně model predikoval kybernetický útok nebo normální chování systému. Ukázka konfúzní matice pro binární klasifikaci je v Obr. 22, kde „True“ znamená kladnou klasifikaci a „False“ zápornou klasifikaci. Reálné třídy v tomto

případě vyjadřují reálnou skutečnost, která je definována v datasetu pomocí tříd. Predikované třídy vyjadřují vytvořené predikce pomocí klasifikačních modelů na základě dat, ale bez znalosti jednotlivých tříd. Detekce anomálií využívá datasetů, ve kterých jsou zastoupena data, spadající pod jednotlivé třídy. V tomto případě jsou kybernetické útoky a normální provoz ICS systému zastoupeny nerovnoměrně. To vyplývá z předpokladu nízkého výskytu anomálií při běžném provozu systému. Tento předpoklad však ovlivňuje výběr metrik pro hodnocení detekčních schopností každého vytvořeného modelu pro detekci anomálií. Například často využívána metrika (Accuracy) vypovídající o přesnosti modelu je v případě detekce anomálií zavádějící z důvodu její závislosti na vyvážeností datasetu z pohledu jednotlivých tříd. Z tohoto důvodu byly voleny metriky hodnocení vhodné pro řešenou problematiku detekce anomálií v oblasti ICS.

V následujícím výčtu jsou popsány metriky, které vycházejí z konfúzní matice a jsou využity v rámci hodnocení prediktivních modelů. Vybrané metriky jsou popsány v řadě publikací [60],[61],[62].

- True negative (TN)…….správně predikované negativní příklady
- True positive (TP)……...správně predikované pozitivní příklady
- False positive (FP)……..špatně predikované pozitivní příklady
- False negative (TN)…….špatně predikované negativní příklady

		Predikované třídy	
		False	**True**
Reálné třídy	**False**	True negative	False positive
	True	False negative	True positive

Obr. 22: Obecná konfúzní matice. [vlastní zdroj]

Následující text popisuje podrobně využité metriky pro hodnocení detekčních schopností každého z algoritmů strojového učení v uskutečněných experimentech.

- **M_{MCC}** – tato metrika (Matthews Correlation Coefficient) je využívána v rámci binární klasifikace, která je charaktistická pro problematiku detekce anomálií. Výsledná hodnota vyjadřuje všechny aspekty konfúzní matice. Není však zatížena nedostatky pří využití nevyváženého datasetu jako tomu je u metriky "Accuracy". M_{MCC} vychází z Pearsonova korelačního

koeficientu, což definuje i její interpretaci. Tato metrika je vyjádřena bezrozměrnou jednotkou, kde její hodnota se pohybuje v rozmezí od $M_{mcc} \in <-1,1>$, kde hodnota -1 představuje nejhorší možný výsledek pro klasifikaci podle tříd a hodnota +1 přestavuje nejlepší možný výsledek pro klasifikaci podle tříd. Přičemž hodnota 0 vyjadřuje zcela náhodný výsledek. [61]

$$M_{MCC} = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (28)$$

- **M_{F1}** – tato metrika (F1 skóre) je dosti využívána pro popis výsledků konfúzní matice. Oproti metrice "Accuracy" není tolik náchylná na výskyt nevyvážených tříd v datasetu. Avšak oproti M_{MCC} nebere v úvahu TN hodnoty v rámci konfúzní matice. Tudíž je více zaměřena na zhodnocení chyb klasifikace modelu. M_{F1} je definována pomocí bezrozměrné jednotky pro rozmezí $M_{F1} \in <0,1>$, přičemž vychází ze dvou metrik M_{Rec} („Recall") a M_{Prec} („Precision") viz. (29, 30). Výsledný vztah pro výpočet metriky M_{F1} je definován v (31). Tuto metriku se snažíme maximalizovat. [61]

$$M_{Rec} = \frac{TP}{TP+FN} \quad (29)$$

$$M_{Prec} = \frac{TP}{TP+FP} \quad (30)$$

$$M_{F1} = \frac{2*M_{Rec} * M_{Prec}}{M_{Rec}+M_{Prec}} \quad (31)$$

- **M_{Prec}** – tato metrika (precision) vypočítává pravděpodobnost pozitivní klasifikace. Vyjadřuje poměr správně pozitivně identifikovaných prvků ke všem prvkům, které jsou označeny jako pozitivní. Tato metodika byla zvolena, protože vyjadřuje vztah mezi TP a FP, přičemž FP vyjadřuje nejdůležitější parametr pro systémy ICS. Falešné poplachy vyjádřené pomocí FP ohrožují ve svém důsledku kontinuitu ICS. Zachování kontinuity procesů je v rámci těchto systémů fundamentálním požadavkem, který musí být za každých okolností naplněn. M_{Prec} je definována pomocí bezrozměrné jednotky pro rozmezí $M_{Prec} \in <0,1>$. Toto kritérium chceme maximalizovat. [60],[62]

$$M_{prec} = \frac{TP}{TP+FP} \quad (32)$$

- **M_{FPR}** – tato metrika (False positive rate (FPR)) vyjadřuje případ, kdy pozitivní třídy jsou identifikovány jako falešné, tedy jedná se o případ, kdy normální a nezávadná komunikace v počítačové síti je vyhodnocena jako

nebezpečná. Na chráněný systém není veden útok, avšak normální komunikace je identifikována jako závadná. Falešné poplachy, jak již bylo v řadě případů vysvětleno v rámci disertační práce, představují zásadní problém pro ICS. Tato metrika je zaměřena na monitorování falešných poplachů v rámci činnosti modelu. M_{FPR} je definována pomocí bezrozměrné jednotky pro rozmezí $M_{FPR} \in <0,1>$. Toto kritérium chceme minimalizovat. [62]

$$M_{FPR} = \frac{FP}{FP+TN} \tag{33}$$

- **Čas** – toto kritérium vyjadřuje čas potřebný k predikci a klasifikaci testovacího datasetu prostřednictvím prediktivního modelu. Čas je v tomto případě parametr, jenž vyjadřuje výpočetní náročnost každého prediktivního modelu. Řada systémů ICS má poměrně dlouhý životní cyklus. Z tohoto důvodu je možné předpokládat nasazení ICS systémů, které nebyly navrhovány s dostatečným výpočetním výkonem z důvodu jejich stáří. Dalším důvodem je vysoká závislost systému ICS na latenci a z toho vyplývá hrozba jejího možného zvýšení prostřednictvím implementace modelu strojového učení. Kritérium času se v tomto případě uvádí v jednotkách sekundy (s).

5.8 INTERPRETACE VÝSLEDKŮ

Interpretace výsledků algoritmů strojového učení je jednou z velmi důležitých oblastí umělé inteligence. Jednotlivé výsledky získané prostřednictvím algoritmů strojového učení mají důležitou hodnotu pro detekci anomálií. Avšak bez interpretace těchto výsledků v rámci kontextu dochází někdy k nesprávné interpretaci. Důležitým prvkem v oblasti identifikace anomálií je i identifikace jejich původu. To usnadňuje následnou analytickou práci při identifikaci zdroje vzniklé anomálie, a tudíž i snižuje čas, při kterém je zasažený systém zranitelný. Řada autorů do nedávné doby přistupovala k algoritmům strojového učení jako k tzv. „černé skříňce“. Jsou známé vstupy a výstupy, ale již je velmi obtížné specifikovat, co vedlo k získaným výsledkům. Z pohledu kybernetické bezpečnosti systémů ICS je porozumění modelům strojového učení nejenom příhodné, ale i zásadní k zajištění bezpečnosti těchto kritických systémů. Autor Ribeiro [63] upozorňuje na často bezbřehou důvěru v prediktivní modely ve velmi kritických oblastech, jako je zdravotnictví nebo detekce teroristů. Mezi tyto oblasti můžeme zařadit i systémy ICS. Interpretace výsledků detekce anomálií je provedena při evaluaci vytvořeného modelu prostřednictvím testovacího datasetu. Hlavním úkolem je vytvoření způsobu ohodnocení významnosti jednotlivých atributů zjištěných anomálií.

6. HLAVNÍ VÝSLEDKY DISERTAČNÍ PRÁCE

V rámci této kapitoly jsou prezentovány jednotlivé výsledky výzkumu v průběhu řešení disertační práce. Chronologicky první částí bezpečnostního výzkumu byla identifikace současných hrozeb a zranitelností v kybernetickém prostoru pomocí kvantitativních a kvalitativních metod. Zde byla zmapována situace a identifikovány potencionálně nejvíce ohrožené oblasti ICS prostřednictvím kybernetických útoků. Na provedený výzkum navazuje analýza s identifikací reálných systémů ICS, které obsahují již známou zranitelnost. Tento postup identifikace potencionálně významných hrozeb v kybernetickém prostředí je nutný pro určení a výběr významných kybernetických útoků, důležitých pro nastavení systému pro detekci anomálií.

Druhá část byla vytvořena z důvodu splnění hlavního cíle disertační práce, který je zaměřen na konceptuální návrh a ověření systému detekce anomálií založeného na strojovém učení v průmyslových řídicích systémech. V rámci řešené problematiky je rozdělen tento bod do několika dílčích celků, v rámci kterých, jsou uskutečněny experimenty. Na jejich základě je v jednotlivých oblastech definována konfigurace systému, která respektuje specifika ICS. Toto řešení je poté ověřeno. Hlavní výsledky disertační práce jsou z velké části založené na velkém počtu experimentů, které zabírají značnou část disertační práce. Z tohoto důvodu byl v této kapitole zvolen následující postup. Velká část podrobných výsledků je uveřejněna v příloze disertační práce. V samotných podkapitolách jsou uvedeny souhrnné výsledky a jejich diskuse.

6.1 IDENTIFIKACE SOUČASNÝCH HROZEB A ZRANITELNOSTÍ ICS V KYBERNETICKÉM PROSTORU

Základním cílem této podkapitoly je provedení výzkumu v oblasti zhodnocení stavu kybernetické bezpečnosti a identifikace nejvíce zranitelných míst pro vedení kybernetického útoku. Jako první oblast zhodnocení zranitelností ICS systémů byla zvolena kvantitativní analýza databáze zranitelností ICS-CERT. Druhým krokem bylo ověření, zdali zranitelnosti v rámci databáze ICS-CERT jsou přítomny v systémech ICS i po svém uveřejnění v databázi. Tento definovaný postup výběru závažných kybernetických útoků je nutný pro validaci a optimalizaci algoritmů strojového učení.

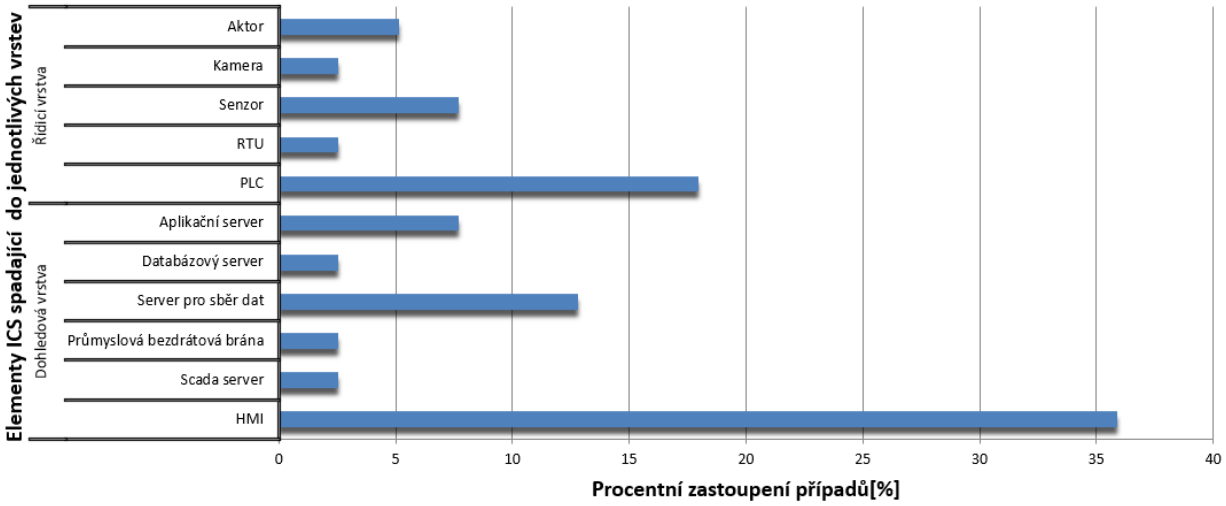
6.1.1 Analýza zranitelností v databázi ICS-CERT

Pro první vyhodnocení zranitelností ICS byla získána a kvantitativně analyzována statistická data z databáze zranitelností ICS-CERT. Získaná data tvoří základní bázi, která je určující pro kvantitativní zhodnocení kybernetické bezpečnosti Řídicí a Dohledové vrstvy. Analyzováno je přibližně padesát zranitelností ve vytyčeném období přibližně šesti měsíců v roce 2015.

Stěžejní otázkou této kapitoly je, která z šetřených hierarchických vrstev představuje větší hrozbu z pohledu kybernetické bezpečnosti. Ta je zkoumána podle následujících hledisek: počet zaznamenaných zranitelností, typu kybernetického útoku v závislosti na nalezené zranitelnosti a závažnosti jednotlivých zranitelností. Na závěr je provedena analýza metriky zneužitelnosti a dopadové metriky.

Pro vyhodnocení hlavního cíle této kapitoly bylo nutné specifikovat jednotlivé vrstvy ICS systému (viz. kapitola 2.1). Ty jsou dále rozčleněny na jednotlivé zájmové prvky (viz. Obr. 23). Analyzovaná data zranitelností jsou řazena do definovaných skupin (viz. Obr. 24). V tomto grafu je znázorněno rozložení zranitelností připadajících na vrstvy, ale také zranitelnosti náležící jednotlivým prvkům definovaného systému.

Lze konstatovat, že nejvíce zranitelností obsahuje Dohledová vrstva s 64% zastoupením z celkových dat. Přičemž nejrizikovějším prvkem je HMI s 36 % celkových zranitelností. Na druhou stranu nelze zanedbávat ani Řídicí vrstvu, na kterou připadá zbývajících 36 % celkových zranitelností. Nejrizikovějším prvkem je PLC s 18 % celkových zranitelností. Zjištěné rozložení dat nás vede k definování dalších cílů výzkumu.

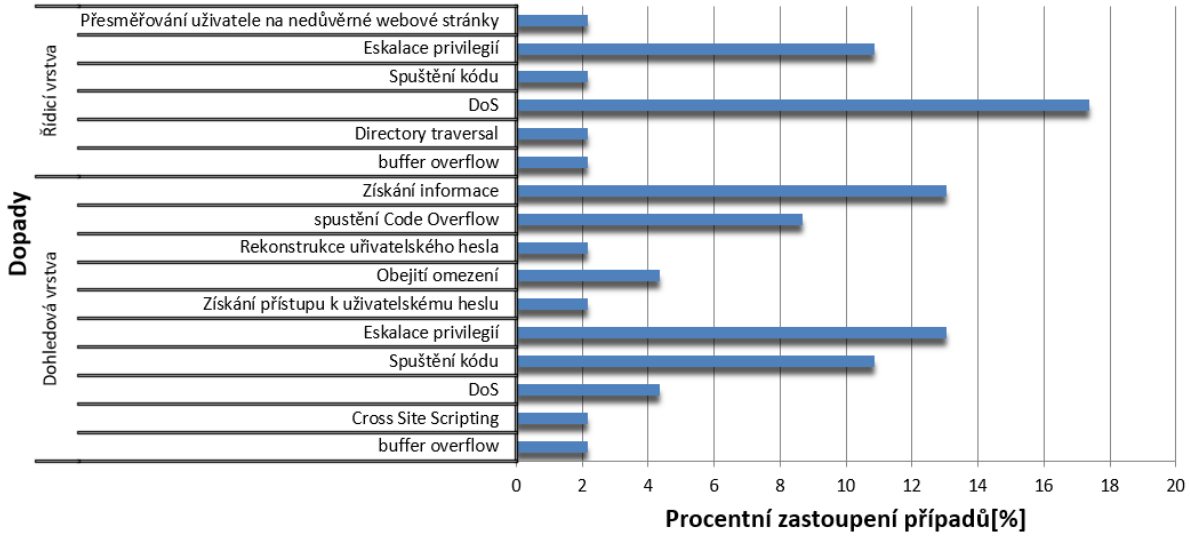


Obr. 23: Rozložení ICS zranitelností. [vlastní zdroj]

Druhým cílem tohoto výzkumu je zmapování možných důsledků vyplývajících ze získaných zranitelností. Rozložení dopadů v závislosti na znalosti zranitelností je graficky znázorněna v Obr. 24, kde jsou jednotlivé dopady rozděleny do jednotlivých vrstev.

Z výsledků vyplývá, že největší procentní podíl dopadu v Dohledové vrstvě zastupuje získání informací s 13 % a eskalace privilegií s 13 %. Můžeme tedy konstatovat, že tato vrstva je velmi zranitelná vůči eskalaci privilegií, které umožnuje útočníkovi průnik do zabezpečeného systému. Dalším

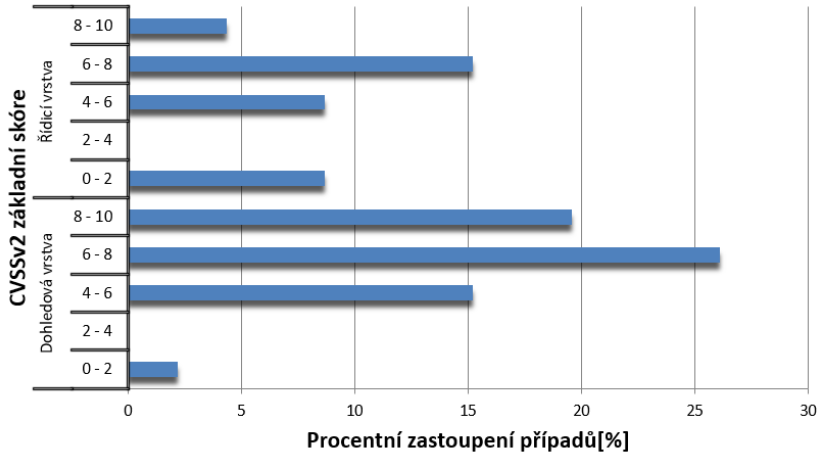
nezanedbatelným dopadem je únik informací zapříčiněný zneužitím zranitelností. V oblasti Řídicí vrstvy je nejvíce zastoupen Denial of Service (DoS) s 17 % a eskalace privilegií s 11 %.



Obr. 24: Rozložení ICS dopadů zranitelností. [vlastní zdroj]

Třetí cíl výzkumu se zabývá procentuálním rozložením závažnosti zkoumaných zranitelností podle standartu pro hodnocení zranitelností Common Vulnerability Scoring System (CVSS) ve verzi 2. Tento systém hodnotí zranitelnosti podle šesti stanovených metrik, které jsou dále využity pro další výzkum viz. Obr. 26 a Obr. 27. Každá zranitelnost je hodnocena na stupnici od nejméně závažné reprezentované 0 až po nejvíce závažnou, která je zastoupena číslicí 10. V rámci této otázky jsou pro jednotlivé vrstvy vytvořeny intervaly, do kterých jsou zařazeny jednotlivé zranitelnosti podle CVSSv2.

Největší počet zranitelností připadá na Dohledovou vrstvu. Avšak z Obr. 25 lze vysledovat, že Dohledová vrstva vyniká také vetší závažností a nebezpečností vyplývající ze zranitelností.



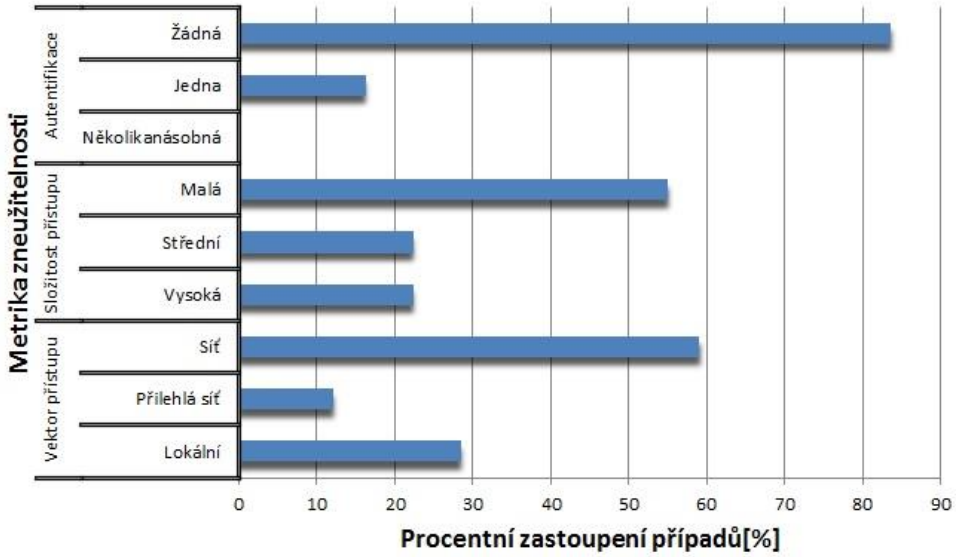
Obr. 25: Specifikace závažnosti analyzovaných zranitelností. [vlastní zdroj]

Následující dva grafy (Obr. 26 a Obr. 27) znázorňují šest metrik používaných pro výpočet CVSSv2. Jedná se o tyto oblasti: Složitost přístupu, Vektor přístupu, Autentifikace, Důvěrnost, Integrita a Dostupnost. Tyto metriky jsou rozděleny do dvou hlavních skupin: Metrika zneužitelnosti a Dopadová metrika. Metrika zneužitelnosti je využita pro analýzu zranitelností (viz. Obr. 26). Zde jsou využity tyto metriky: Složitost přístupu, Vektor přístupu, Autentifikace. Ty umožnují definovat a ocenit každou zranitelnost podle přesně určených kritérií.

Metrika Složitosti přístupu charakterizuje míru speciálních podmínek, které musí nastat, aby bylo možné využít zjištěnou zranitelnost. Z tohoto důvodu je metrika Složitosti přístupu rozdělena do tří oblastí podle míry počátečních podmínek. Z analyzovaných dat vyplývá, že značná část zranitelností (55 %) nepotřebuje k jejich zneužití další speciální podmínky viz. Obr. 26.

Vektor přístupu je druhá metrika určená ke zhodnocení, jakého přístupu musí útočník dosáhnout, aby zneužil nalezenou zranitelnost. Podle přístupu útočníka je tato metrika rozdělena do tří oblastí. Aby útočník využil zranitelnost, musí mít lokální přístup do fyzického systému (lokální), přístup do sítě, ve které se nachází systém se zranitelností (přilehlá síť) nebo nemusí mít fyzický přístup do lokální sítě (síť), což vede ke vzdálenému zneužití. Z výsledků vyplývá, že největší počet analyzovaných zranitelností lze zneužít vzdáleně 59 %.

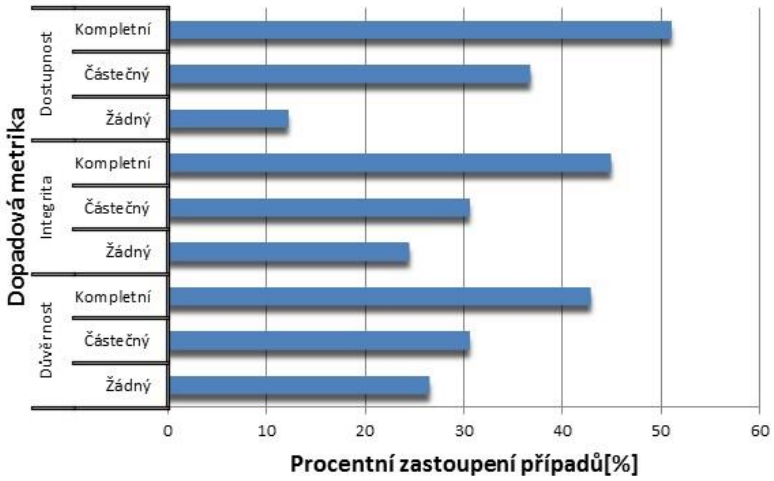
Další metrika, kterou byla zkoumaná data podrobena, je Autentifikace. Ta rozděluje zranitelnosti podle toho, jestli k jejich zneužití je potřeba provést autentizaci jednou, vícekrát nebo není potřeba provést ani jednou. Největší zastoupení mezi analyzovanými zranitelnostmi mají ty, které lze zneužít bez nutnosti autentizace s 84 %.



Obr. 26: Specifikace ICS zranitelností na základě Metriky zneužitelnosti. [vlastní zdroj]

Závěrečná část této kapitoly je zaměřena na specifikaci dopadů vycházejících z analyzovaných dat. Tuto problematiku popisuje skupina známá jako Dopadová metrika. Zde jsou dopady rozděleny do tří podskupin, které představují jednotlivé metriky: Integrita, Důvěrnost, Dostupnost. Každá ze zranitelností je definována v rámci jednotlivých metrik, přičemž je řešen žádný, částečný nebo úplný dopad na Integritu, Důvěrnost, Dostupnost zasaženého systému.

Z vyhodnocených dat vyplývá, že největší dopad byl zaznamenán v rámci metodiky Dostupnost. Zkoumané zranitelnosti mohou s největší pravděpodobností zapříčinit kompletní ztrátu dostupnosti systému. Celkem se jedná o 51 % ze všech zranitelností. I v ostatních oblastech je nejvíce zastoupena možnost kompletní ztráty Důvěrnosti (43 %) a Integrity (45 %).



Obr. 27: Specifikace ICS zranitelností na základě Dopadové metriky. [vlastní zdroj]

Dílčí závěr

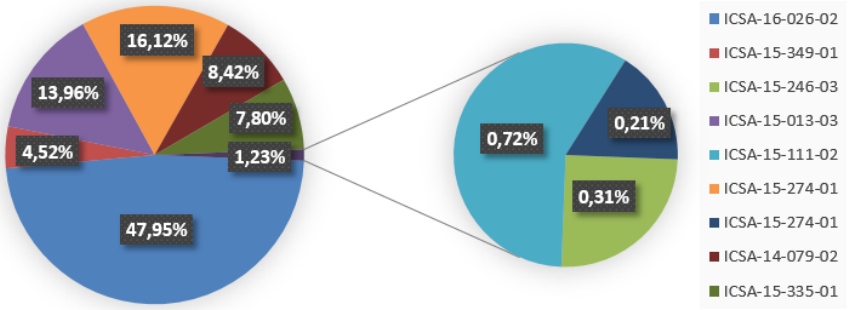
Z výzkumu vyplývá, že Dohledová vrstva je zatížena vyšším výskytem zranitelností, které generují závažnější dopady než v Řídicí vrstvě. Ty mohou být zneužity k průniku do systému ICS a získání důležitých informací. Je nutné však upozornit i na zvýšený výskyt zranitelností, které mohou vést k DoS. Ten představuje závažnou hrozbu dostupnosti služeb, která je nejdůležitějším bezpečnostním kritériem pro ICS. Tuto tezi podporují informace vyplývající z analýzy Dopadové metriky. Z ní je zřejmé, že analyzované zranitelnosti mají největší dopad právě na dostupnost. Tuto skutečnost umocňuje ještě fakt, že většinu z analyzovaných zranitelností lze vzdáleně zneužít, přičemž ve většině případů k tomu není potřeba žádné autentizace.

6.1.2 Vyhledání reálných systémů ICS se zjištěnou zranitelností

Prvním cílem výzkumu bylo zhodnocení současného stavu ICS prvků přístupných prostřednictvím internetového připojení. Pro naplnění tohoto cíle byl využit Shodan nástroj pro detekci a identifikaci internetově připojených zařízení

a ICS-CERT databáze zranitelností, která poskytuje aktuální seznam ICS zranitelností. Hlavním předpokladem této části byla časová prodleva mezi zveřejněním zranitelnosti v databázi ICS-CERT a reálnou aktualizací systému z důvodu odstranění zranitelnosti. Tato zranitelnost byla umocněna faktem, že samotné aktualizace jsou pro ICS, a tudíž i pro SCADA systémy velmi kritické, a proto je nelze provádět na denní bázi z důvodu jejich testování. Proto byl zmiňovaný interval mezi zveřejněním zranitelnosti a jejím odstraněním poměrně dlouhý. Pro identifikaci ICS zranitelnosti byla využita databáze ICS-CERT. Dalším kritériem pro výběr zranitelnosti byla otázka otevřenosti systému především prostřednictvím Internetu. ICS systémy se zranitelnostmi, pro které byla uskupením ICS-CERT nařízena absolutní izolace od Internetu z důvodu mitigačních opatření se stala zájmovou skupinou pro zvolený výzkum. Takto charakterizované zranitelnosti byly nejprve získány z databáze ICS-CERT a poté následně analyzovány. Všechna potřebná data, týkající se reálných ICS systémů, byla zajištěna prostřednictvím nástroje SHODAN a metody "Banner grabbing". Bylo shromážděno poměrné velké množství zranitelných zařízení.

V roce 2016 bylo shromážděno 974 zranitelných ICS systémů. Ty i přes doporučení organizací ICS-CERT na jejich izolaci od internetového připojení, byly reálně dosažitelné právě prostřednictvím internetového připojení. Výsledky jsou znázorněny v Obr. 28.



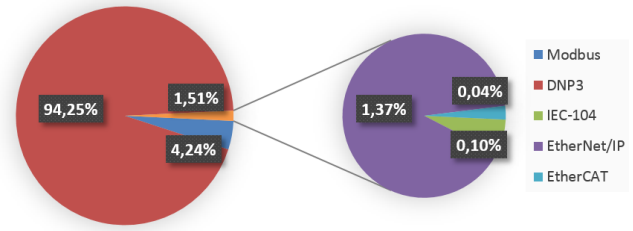
Obr. 28: Rozpis ICS systémů na jednotlivé zranitelnosti. [vlastní zdroj]

Nejvíce zasaženou zemí byly Spojené státy americké, kde bylo 487 zranitelných systémů. Na druhém místě bylo Španělsko se 75 zranitelnými systémy a na třetím místě Kanada s 59 zranitelnými systémy. Z výsledků vyplývá, že na celou Evropu připadá 291 zranitelných systémů, což nepřesahuje celkový počet postižených systémů připadající na USA. Lze také konstatovat, že téměř 50 % všech zasažených systémů bylo zasaženo zranitelností popsanou jako ICSA-16-026-02.

Komparace průmyslových komunikačních protokolů z bezpečnostního hlediska

Druhým cílem této kapitoly bylo specifikovat zranitelnosti systémů využívající průmyslové komunikační protokoly v návaznosti na operační systémy. Bylo analyzováno pět průmyslových protokolů. Ty jsou hlavními a běžně využívanými reprezentanty průmyslových komunikačních protokolů. Jedná se o: Modbus – port 502, DNP3 – port 20000, IEC-104 – port 2404, EtherNet/IP – port 44818, EtherCAT – port 34980. Identifikovány byly ICS systémy, které využívají zmíněné průmyslové komunikační protokoly a zároveň dnes již zranitelný operační systém Windows XP.

Podle výsledků této analýzy bylo shromážděno 317 891 internetově připojených ICS systémů využívajících jeden z výše vyjmenovaných průmyslových komunikačních protokolů. Rozpis shromážděných ICS systémů podle průmyslových komunikačních protokolů je znázorněn v Obr. 29.



Obr. 29: Rozpis shromážděných ICS systémů. [vlastní zdroj]

Z výsledku vyplývá, že 94,25 % testovaných ICS systémů je provozováno s průmyslovým komunikačním protokolem DNP3. Takové široké zastoupení zařízení na Internetu může vest ke značnému rozsahu kybernetických útoků vedených vůči takto specifikovaným systémům. V druhé části tohoto bodu byly identifikovány systémy a podrobeny dalšímu zkoumání. Dále byly selektovány systémy, které využívaly operační systém Windows XP, jejichž kybernetická bezpečnost je v dnešní době sporná. Ze shromážděných dat lze konstatovat, že 188 systémů disponuje Windows XP, přičemž 132 systémů připadá na DNP3 průmyslový komunikační protokol a 56 systémů připadá na Modbus průmyslový komunikační protokol. Z výsledků lze konstatovat, že ICS systémy využívající průmyslový komunikační protokol DNP3 jsou zranitelnější vůči identifikaci přes internetové připojení nástroji jako je Shodan.

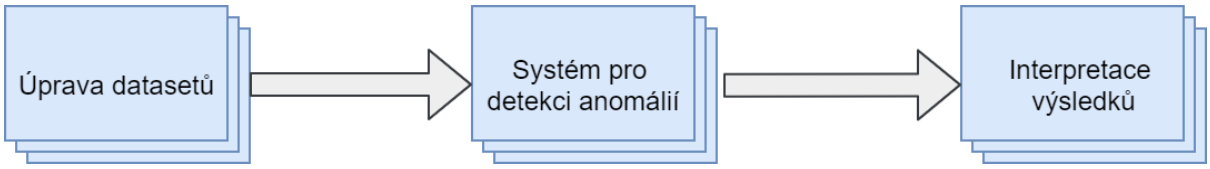
Dílčí závěr

Tato kapitola byla zaměřena na provedení analýz ve vztahu k výskytu zranitelností pro prostředí ICS. Z výsledků vyplývá značný výskyt zranitelností ovlivňující všechny oblasti metriky: dostupnost, důvěrnost a integrita. Druhá část byla zaměřena na nalezení takto identifikovaných zařízení ICS. Z výsledků

vyplývá velmi vysoká míra zranitelných ICS zařízení dostupných pomocí internetového připojení. To je zejména z důvodu pomalého procesu aktualizace v oblasti ICS. Identifikace hrozeb v rámci kybernetické bezpečnosti umožňuje lepší pochopení a správné směřování kybernetické ochrany. Takto zjištěné informace mohou být využity pro testování a nastavení systému detekce anomálií pro zajištění efektivnější kybernetické ochrany.

6.2 KONCEPTUÁLNÍ NÁVRH A OVĚŘENÍ SYSTÉMU DETEKCE ANOMÁLIÍ V PRŮMYSLOVÝCH ŘÍDICÍCH SYSTÉMECH

Hlavním cílem disertační práce je konceptuální návrh a ověření systému detekce anomálií, vztahujícího se a respektujícího specifika průmyslových řídicích systémů. Z tohoto důvodu je tato hlavní část výsledků disertační práce rozdělena do tří podsekcí, které reflektují dílčí postupy při naplnění cílů disertační práce. Základní postup je znázorněn v diagramu zobrazeném na Obr. 30. Tento diagram slouží jako základní orientační bod v rámci představeného výzkumu. Jeho modifikace byla využita pro zmapování postupu následujícími podkapitolami.

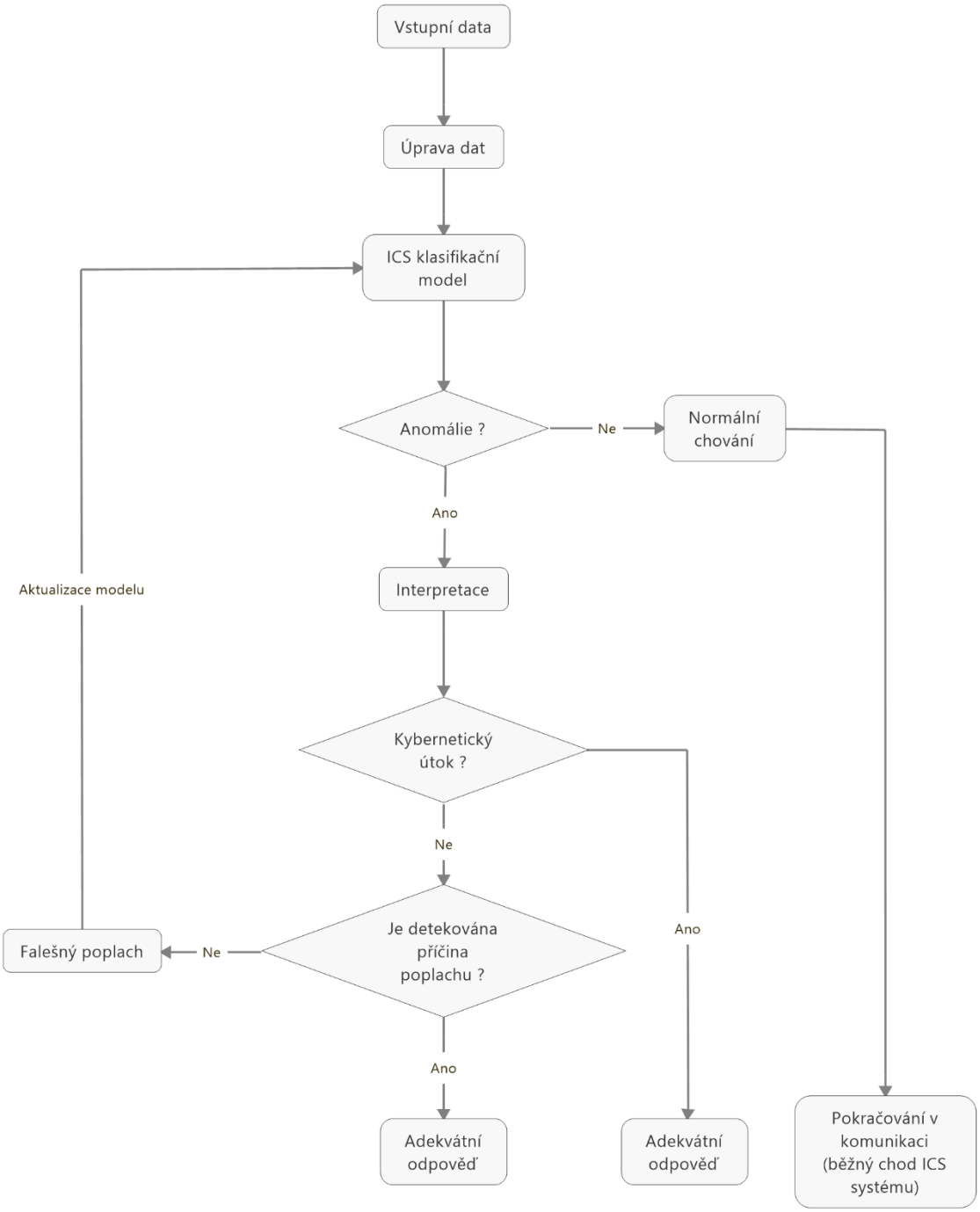


Obr. 30: Diagram procesů pro tvorbu algoritmu. [vlastní zdroj]

V rámci první podsekce jsou definovány procesy nezbytné pro nasazení jakéhokoliv algoritmu pro detekci anomálií. Úprava vstupních dat (trénovacího datasetu, validačního datasetu a testovacího datasetu) je jedním z podstatných kroků pro tvorbu efektivního systému pro detekci anomálií.

Druhá podsekce této kapitoly je zaměřena na algoritmy strojového učení a optimalizační techniky ve spojitosti s metodou multikriteriálního rozhodování (TOPSIS). Zde jsou popsány procesy systému detekce anomálií. Tato podkapitola by se dala také označit jako nejobsáhlejší podkapitola. Její obsáhlost totiž koresponduje s její významností, jelikož tvoří hlavní část jádra disertační práce. Výsledný systém detekce je ověřen za využití kybernetických útoků, které nebyly využity pro tvorbu systému detekce anomálií. V poslední a závěrečné podkapitole jsou definovány procesy pro interpretaci výsledků získaných prostřednictvím systému pro detekci kybernetických útoků.

Důležitou stránkou výzkumu, zejména pro praxi, je návrh provozu systému detekce anomálií v běžném provozu. Tento konceptuální návrh pro využití představeného algoritmu pro detekci anomálií je znázorněn v Obr. 31.

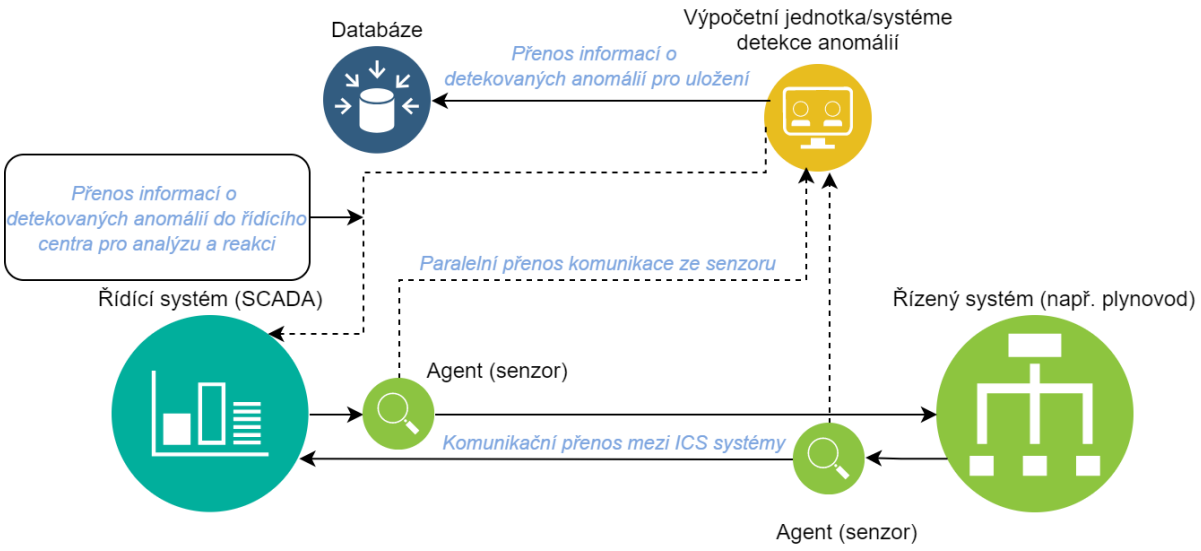


Obr. 31: Konceptuální návrh systému pro detekci anomálií. [vlastní zdroj]

Vstupní data ze systému v reálném čase musí být upravena na stejný formát, jaký byl využit k vytvoření prediktivního (klasifikačního) modelu. Tudíž trénovací a testovací dataset musí mít stejný formát. Pro vstupní data jsou predikovány data podle vytvořeného modelu. Následně je vyhledávána statistická odchylka od optimálního stavu podle prediktivního modelu. Periodická aktualizace klasifikačního modelu je nutná pro zachování jeho relevantnosti. Jestliže je nalezena odchylka (anomálie) od predikovaného chování je nutné rozlišit, jestli se jedná o kybernetický útok. Pomocí interpretace (více v kapitole

6.2.4) a další odborné analýzy je rozhodnuto, jestli se jedná o kybernetický útok, nebo jestli má anomálie jinou příčinu (např. výpadek nebo porucha systému). Není-li nalezena příčina, pak se má za to, že klasifikační model neodpovídá realitě, a je nutné ho aktualizovat. Jinak by docházelo k falešným identifikacím kybernetických útoků. V případě nenalezení anomálie je pokračováno v normálním fungování ICS systému. Tedy přenosu informací mezi řídicím systémem a řízeným systémem.

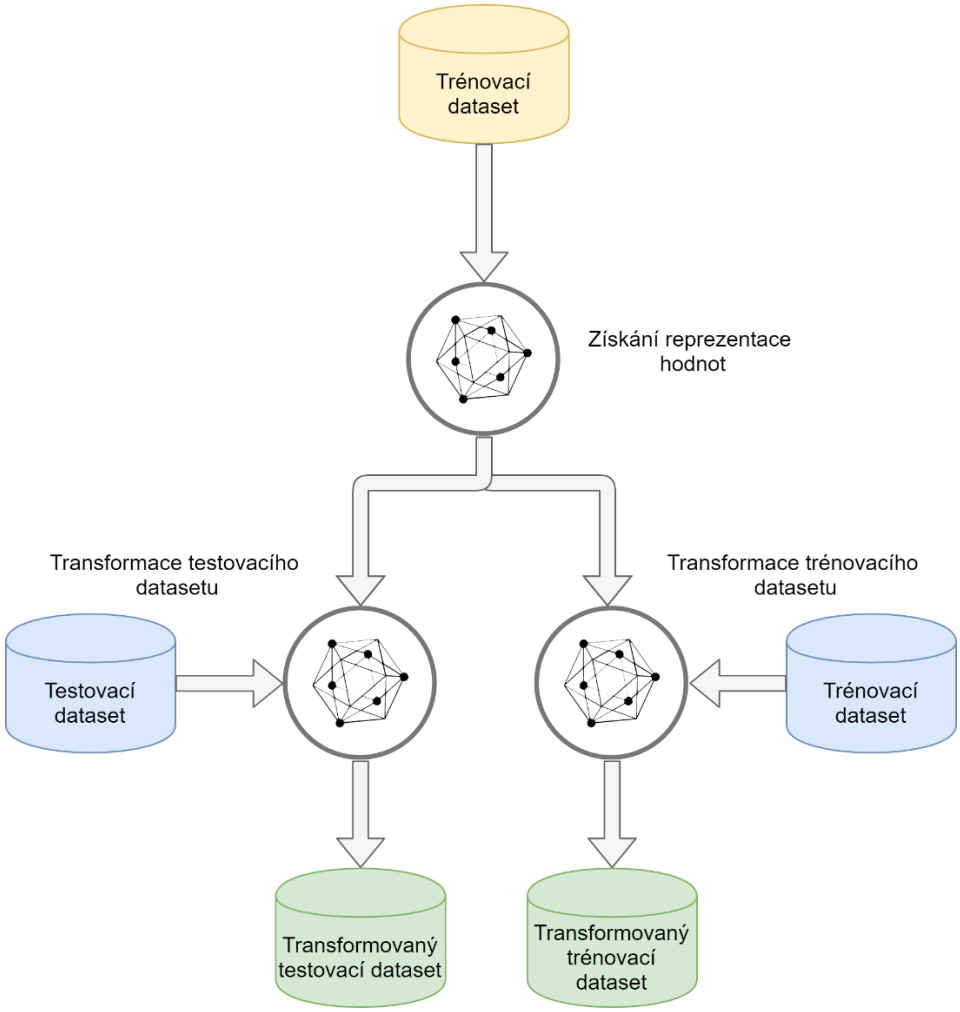
Otázka implementace systému detekce anomálií v reálném provozu je jednou z důležitých problematik v rámci disertační práce. Schématický diagram zobrazený v Obr. 32 představuje implementaci systému pro detekci anomálií v prostředí ICS. Představený systém je fundamentálně založený na datech z chráněného systému. Na základě těchto dat je vytvořen klasifikační model, který zajišťuje detekci anomálií. Z tohoto důvodu je nutná implementace agentů (senzorů) mezi řízeným a řídicím systémem pro snímání veškeré komunikace mezi těmito systémy. Paralelní záznam této komunikace je poté zasílán do výpočetní jednotky, kde je umístěn systém pro detekci anomálií. Na základě komparace reálné komunikace s vytvořeným modelem je identifikováno anomální chování. O tomto chování by měl být veden záznam v databázi pro pozdější analýzu. Zároveň s tímto přenosem jsou potřebné informace zaslány do řídicího střediska, kde je provedena počáteční analýza situace a přijata nezbytná opatření na základě aktuální situace.



Obr. 32: Implementace systému detekce anomálií v systému ICS. [vlastní zdroj]

Všechny transformace v provedených experimentech pro oblast detekce anomálií musí vycházet z předpokladu neznalosti testovacích dat. K tomu byla využita třída „pipeline“ spadající do knihovny sklearn pro programovací jazyk Python. Pomocí této knihovny je každá z využitých transformací nastavena pomocí trénovacího datasetu, tudíž výsledné transformace nevyužívají data, která by mohla být kontaminována kybernetickým útokem. Diagram, znázorněný

na Obr. 33, znázorňuje využití třídy „pipeline“ v případě normalizace datasetu v rozmezí mezi hodnotami 0 a 1. V diagramu je názorně popsán celý proces transformace. Reprezentace hodnot pro normalizaci v rozpětí 0 a 1 je získána z trénovacího datasetu. Tato reprezentace je dále využita pro transformaci trénovacího a testovacího datasetu. Pokud se v rámci testovacího datasetu objeví unikátní hodnota pro daný atribut – anomálie, pak je tato hodnota transformována podle získané reprezentace. Tudíž může nabývat i vyšších hodnot, než je rozmezí od 0 po 1. Takto transformovaná data neztrácejí informační hodnotu, a tedy podporují detekci anomálií v testovacím datasetu.



Obr. 33: Využití třídy „pipeline“ pro transformaci datasetu. [vlastní zdroj]

V rámci řešené problematiky bylo využito programovacího jazyka Python pro úpravu datasetů, tvorbu a testování modelů strojového učení, interpretaci výsledků. Zvláště bylo využito knihoven **Keras**, **TensorFlow** a **Scikit-learn**. Práce byla podpořena prostředky **A.I.Lab** na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně (ailab.fai.utb.cz). Také bych rád zmínil autora Matt Harvey, který mi pomohl s aplikací evolučního algoritmu. A v neposlední řadě je velice oceňován přístup k výpočetním a skladovacím zařízením vlastněnými stranami a projekty přispívajícími do národní sítě gridové

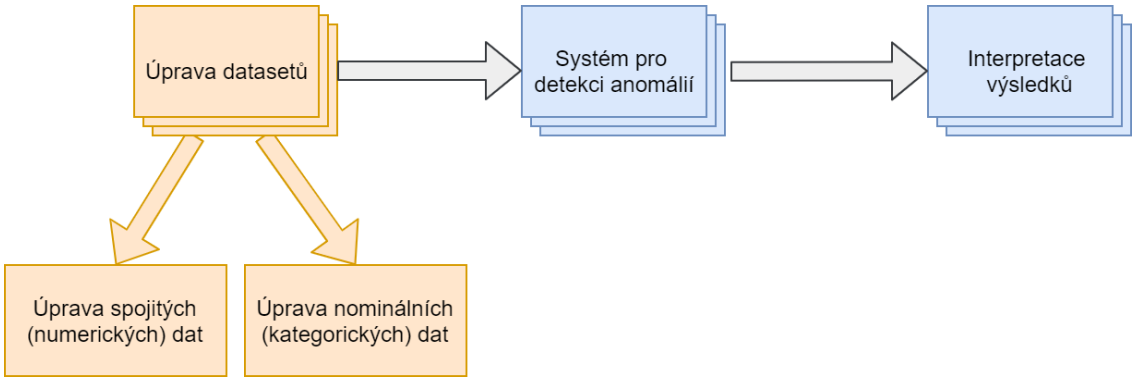
infrastruktury **MetaCentrum** poskytované v rámci programu „Projekty velkých infrastruktur výzkumu, vývoje a inovací“ (CESNET LM2015042), které přispěly k dokončení výzkumu a tím i této disertační práce.

Omezení výzkumu v disertační práci

Disertační práce je zaměřena na detekce anomálií v oblasti kybernetické bezpečnosti. Jejím hlavním cílem je tvorba a testování systému detekce anomálií, především s přihlédnutím k reálnému využití v rámci systému ICS. Z tohoto důvodu lze provedený výzkum zařadit do aplikační oblasti. Provedený výzkum není koncipován s ambicemi v oblasti základního výzkumu. Tento postup je zvolen z důvodu poměrně velkého rozsahu samotné disertační práce. Řada využitých technik, algoritmů či postupů je poměrně známa a již zpracována. Z tohoto pohledu je vhodné využít zdrojů, které jsou veřejně přístupné a akceptované vědeckou komunitou. Disertační práce je zaměřena především na vytvoření uceleného systému obsahujícího identifikaci potencionálních kybernetických hrozeb, detekci kybernetických útoků v podobě anomálií a jejich interpretaci pro specifickou oblast ICS systémů.

6.2.1 Úprava datasetů

V rámci této podkapitoly jsou prezentovány výsledky řady experimentů. Na jejich základě byly zvoleny optimální postupy a techniky pro úpravu datasetů z pohledu kybernetické bezpečnosti ICS. Vybraná řešení byla otestována pro rozdílné algoritmy strojového učení. V rámci každého z vybraných datasetů je řešena problematika chybějících hodnot, normalizace datasetů, transformace dat do příhodné podoby (nominální data, ordinální data atd.). Tato podkapitola je rozdělena na dvě fundamentálně odlišné části (viz. Obr. 34).



Obr. 34: Diagram procesů pro tvorbu algoritmu – úprava datasetů. [vlastní zdroj]

První z nich byla zaměřena na úpravu numerických spojitých hodnot do vhodného tvaru pomocí vybraných metod normalizace, standardizace, náhrady chybějících hodnot. V rámci této skupiny byly provedeny experimenty, které podporují zvolené řešení pro další využití v rámci navazujících kapitol. Druhá

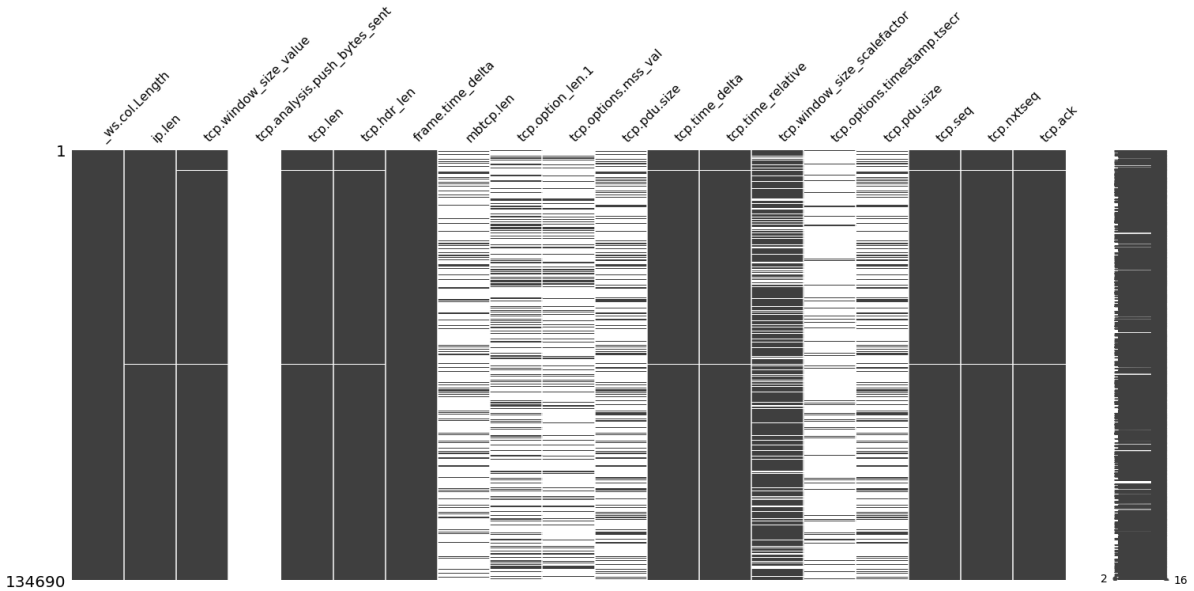
část je zaměřena na popis zvoleného řešení pro zpracování kategorických dat, která mají nominální charakter.

Atributy síťové komunikace mohou být rozčleněny do dvou skupin, které jsou reprezentovány rozdílnými datovými typy. V rámci této podkapitoly jsou definovány postupy pro úpravu spojitých hodnot numerické povahy a kategorických dat. Numerické atributy reprezentují poměrně menší skupinu atributů oproti druhé skupině, do které řadíme nominální hodnoty v podobě kategorických dat.

Formát dat získaných z reálných systémů je obvykle nekompatibilní s převážnou částí algoritmů strojového učení. Tato data jsou v řadě případů neúplná a nekonzistentní. Techniky pro úpravu datasetů umožňují, aby je algoritmy strojového učení zpracovaly pro vytvoření prediktivních modelů. Navíc tyto techniky obvykle zvyšují přesnost predikce modelu. Z těchto důvodu je nutné vstupní data upravit a transformovat do správného formátu.

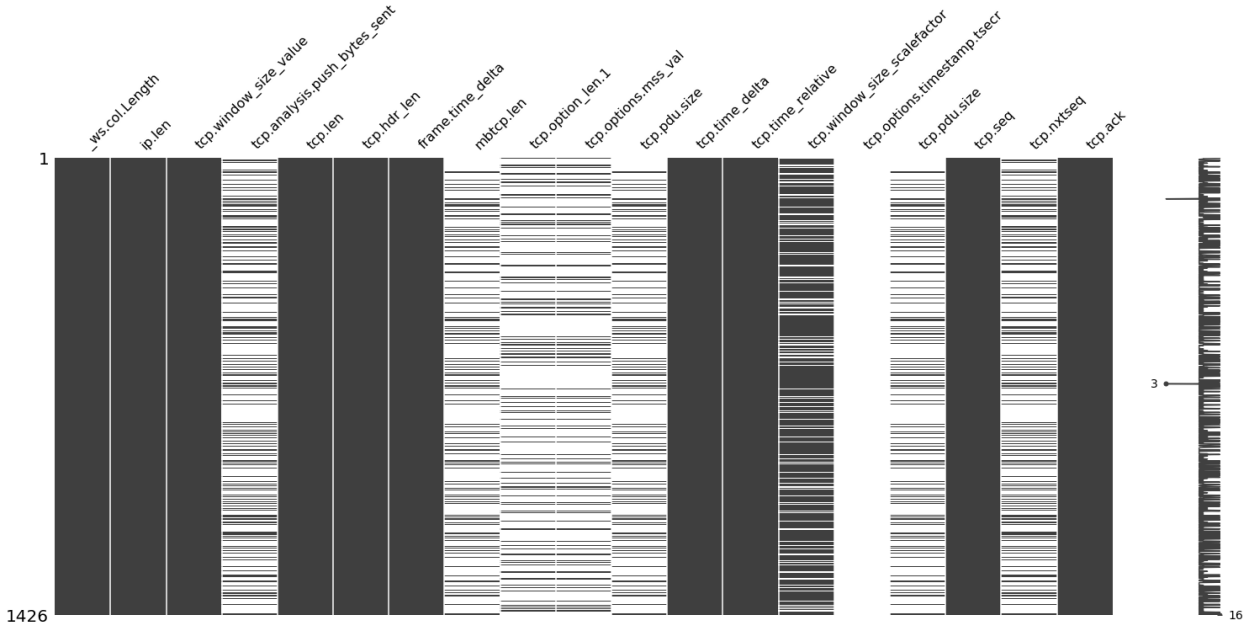
V rámci této sekce byly provedeny experimenty, jejichž účelem bylo definovat nejvhodnější variantu úpravy vstupních dat v rámci předkládaného systému pro detekci anomálií, a to jak z hlediska numerických dat, tak z hlediska nominálních dat. Pro experiment byl vybrán dataset 1 (ICS Modbus dataset). Ten byl využit pro trénování a testování modelů strojového učení.

Výběr atribut datasetu v rámci řešeného problému detekce anomálií může být poměrně riskantní. Existuje množství technik pro výběr atributů podle jejich významnosti. Avšak tyto techniky potřebují plně označená data s definovanými třídami (pro každý z datasetů). Z tohoto pohledu je výběr atributů v rámci problematiky detekce anomálií velmi obtížný proces. Tuto skutečnost lze demonstrovat na základě vizualizace trénovacích a testovacích dat v rámci datasetu 1 (viz. Obr. 35 a Obr. 36).



Obr. 35: Reprezentace chybějících hodnot v trénovacím datasetu (dataset 1). [vlastní zdroj]

Na Obr. 35 je patrný výskyt chybějících hodnot v rámci trénovacího datasetu. Chybějící hodnoty jsou reprezentovány černou vodorovnou čárou v rámci každého atributu. Lze vidět, že atribut "tcp.analysis.push_bytes_sent", je prázdným atributem. V tomto případě by většina technik pro selekci a výběr atributů označila tento atribut jako nevýznamný, proto by byl vyřazen z trénovacího datasetu.



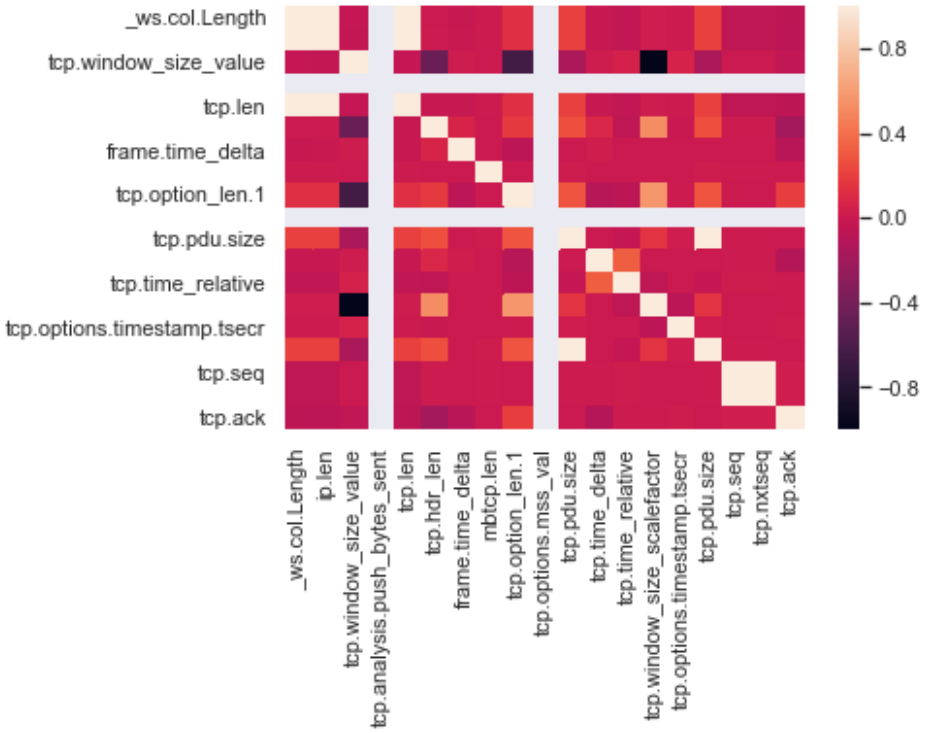
Obr. 36: Reprezentace chybějících hodnot v testovacím datasetu (dataset 1). [vlastní zdroj]

Avšak, jak je možno vidět v Obr. 36, v rámci testovacího datasetu (obsahující kybernetický útok – CA1_1) již tento atribut obsahuje reálné hodnoty. Z tohoto důvodu je v rámci disertační práce využito celého spektra atributů pro prezentované experimenty. Takový postup však vede k vysoké dimensionalitě datasetu. Tento problém je často řešen pomocí technik pro redukci atributů.

Numerická data

V rámci numerických dat jsou provedeny dvě transformace. První je zaměřena na náhradu chybějících hodnot v rámci datasetu. Je využito náhrady pomocí aritmetického průměru, mediánem a konstantou. Druhá skupina transformací je zaměřena na normalizaci nebo standardizaci numerických hodnot.

V rámci analýzy atributů využívaného datasetu byla proveden analýza korelací mezi jednotlivými atributy trénovacího datasetu. Výsledky jsou zobrazeny v diagramu „heatmap“ na Obr. 37. Z provedené analýzy vyplývá velmi slabý lineární vztah mezi jednotlivými atributy. Tento fakt by mohl znamenat problémy pro některé algoritmy strojového učení. Avšak v tomto případě je nutné mít na paměti, že pracujeme jen s menší částí datasetu, a tudíž pro výslednou detekci anomálií je využita větší množina atributů.



Obr. 37: Diagram Heatmap pro atributy v trénovacím datasetu (dataset 1). [vlastní zdroj]

Diagram postupu úpravy numerických hodnot pro vybrané atributy z datasetu 1 je zobrazen v Obr. 38. Tento diagram se skládá ze tří částí (tabulek). Každá

z nich obsahuje pět atributů. Jednotlivé tabulky představují dílčí části procesu úpravy datasetu.

1. První tabulka představuje data v tzv. „syrovém stavu“. V tomto případě byla tato data zaznamenána pomocí senzorů ve sledovaném systému.
2. Druhá tabulka představuje data, která jsou očištěna od nulových hodnot. V tomto případě prostřednictvím aritmetického průměru.
3. Poslední tabulka představuje data, která prošla procesem normalizace. V tomto případě v rozmezí <-1,1>.

tcp.len	tcp.hdr_len	frame.time_delta	mbtcp.len	tcp.option_len.1
0	28	0		4
0	28	0.000185		4
0	20	0.000155		
12	20	0.00056	6	
0	28	0.000589		4
0	28	0.000173		4
17	20	9.40E-05	11	
0	20	7.40E-05		
12	20	0.000547	6	

Náhrada nulových hodnot

tcp.len	tcp.hdr_len	frame.time_delta	mbtcp.len	tcp.option_len.1
0	28	0	6.169955	4
0	28	0.000185	6.169955	4
0	20	0.000155	6.169955	5.85009
12	20	0.00056	6	5.85009
0	28	0.000589	6.169955	4
0	28	0.000173	6.169955	4
17	20	9.40E-05	11	5.85009
0	20	7.40E-05	6.169955	5.85009
12	20	0.000547	6	5.85009
0	28	0.000546	6.169955	4

Normalizace

tcp.len	tcp.hdr_len	frame.time_delta	mbtcp.len	tcp.option_len.1
0	0.333333	0	0.309994	0
0	0.333333	2.72E-05	0.309994	0
0	0	2.28E-05	0.309994	0.308348
0.008219	0	8.22E-05	0.285714	0.308348
0	0.333333	8.65E-05	0.309994	0
0	0.333333	2.54E-05	0.309994	0
0.011644	0	1.38E-05	1	0.308348
0	0	1.09E-05	0.309994	0.308348
0.008219	0	8.03E-05	0.285714	0.308348
0	0.333333	8.02E-05	0.309994	0

Obr. 38: Proces úpravy numerických hodnot datasetu pro vybrané atributy (dataset 1). [vlastní zdroj]

Nominální data

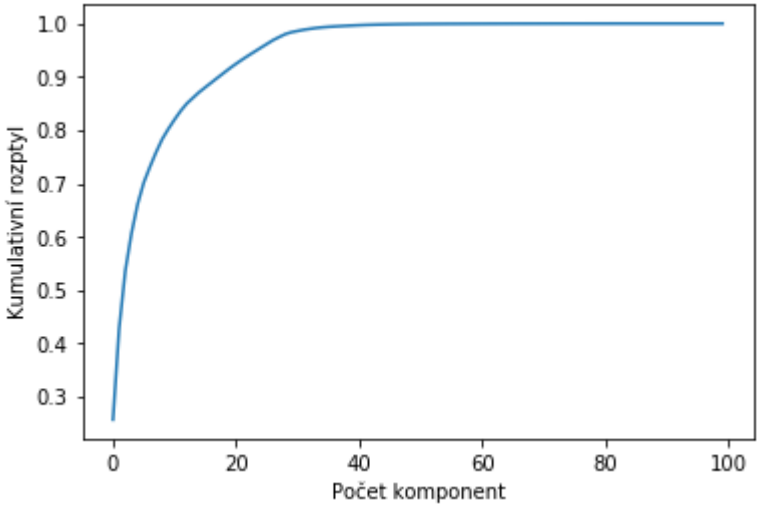
Nominální data představují v rámci řešené problematiky proporcionálně rozsáhlejší skupinu atributů oproti numerickým atributům. Z tohoto důvodu je zásadní zvolit vhodný postup úpravy datasetů. Ve spojitosti s úpravou nominálních dat byly využity dvě transformace, které umožňují racionální definice vztahů mezi nominálními daty při zachování možnosti zpracování rozsáhlých datasetů. Zvolené řešení bylo vybráno i s ohledem na interpretaci výsledků.

Každý z nominálních atributů je v první fázi identifikován. Posléze jsou tyto vybrané atributy převedeny do binární podoby. Tento postup je zvolen z důvodu tvorby datasetu, ve kterém by jednotlivé hodnoty v rámci jednoho atributu měly definovanou stejnou váhu. Vybraný postup je však velmi náročný z důvodu velké dimenze nového datasetu. Každá unikátní hodnota v každém z nominálních atributů je transformována do nového atributu, který nabývá binární podoby. Výsledná transformační tabulka je vytvořena na základě trénovacího datasetu. Pro nové unikátní hodnoty, které se mohou vyskytnout v testovacím datasetu je vytvořen nový atribut pro každý z původních atributů. Tyto atributy reflektují výskyt nových, dosud neznámých hodnot, které jsou přítomny v testovacím datasetu.

Výsledný dataset (trénovací, validační nebo testovací) je ve většině případů zatížen velmi vysokou dimenzí. Takový dataset může zapříčinit zvýšení výpočetní náročnosti zvoleného řešení až za pomyslnou mez, za kterou by nebylo možné navrhovaný detekční systém využít v reálném prostředí. Z tohoto důvodu bylo nutné využít technik pro redukci dimenze datasetu. Pro splnění tohoto úkolu byl vybrán algoritmus PCA, který byl detailněji popsán v kapitole 4.2.1. Tento algoritmus vytváří nový dataset o nižší dimenzi atributů, avšak při zachování velké části informací původního datasetu. Nové atributy se nazývají komponentami. Ty vznikají kombinací původních atributů.

Pro sestavení hlavních komponent je nutné nejprve určit jejich počet. Určení tohoto počtu lze provést za pomocí kumulativního rozptylu. PCA algoritmus se snaží konvertovat co možná nejvíce informací z atributů do první komponenty, dokud nebude moci pojmout další informace. Tento proces se iterativně opakuje, přičemž každou další komponentou pojme méně informací. Tento postup se opakuje do té doby, dokud další komponentou již nenavýšíme uchované informace v rámci nového datasetu. Na Obr. 39 je vyobrazen kumulativní rozptyl pro vytvořené komponenty (100 komponent). Z popisovaného obrázku lze pozorovat trend, při kterém nejvíce rozptylu připadá na první komponenty. Tento proces se ustaluje přibližně v 35 komponentě. Výběr počtu komponent pro PCA lze určit prostřednictvím dvou základních způsobů. Prvním z nich je expertní

odhad podle vytvořeného grafu v Obr. 39. Tento postup však vyžaduje intervenci lidského faktoru. Druhý způsob výběru je založen na definování míry rozptylu, jenž by měl být zahrnut v novém datasetu. V rámci disertační práce byla zvolena druhá možnost z popisovaných. Hranice pro definování množství komponent v rámci PCA byla zvolena pro všechny experimenty stejně. Nový dataset vždy představuje 99 % rozptylu původního datasetu.



Obr. 39: Vývoj kumulativního rozptylu v závislosti na počtu komponent (dataset 1). [vlastní zdroj]

Postup transformace nominálních dat je demonstrován v Obr. 40. První tabulka diagramu představuje tři atributy (zdrojovou a cílovou IP adresu a použitý protokol). Jedná se jen o dílčí výřez použitých atributů. Každý z těchto vybraných atributů je převeden do binární podoby. Tedy pro každou unikátní hodnotu v rámci atributu je vytvořen nový atribut. Takto vytvořené binární atributy nabývají hodnot 0 a 1 (0 – hodnota není přítomna v rámci sledovaného datového bodu, 1 – hodnota je přítomna v rámci sledovaného datového bodu). Transformovaná data jsou zobrazena v rámci druhé tabulky v Obr. 40. Z důvodu vysoké dimenze transformovaného datasetu, který nabýval 76 dimenzí, bylo zvoleno zobrazení pouze 8 dimenzí. Poslední transformace v rámci nominálních dat je založena na algoritmu PCA. Byly vytvořeny hlavní komponenty, které reprezentují původní dataset. Původních 76 dimenzí bylo redukováno na 24 dimenzí, při zachování 99 % rozptylu původního datasetu. Jednotlivé hlavní komponenty jsou zobrazeny také na Obr. 40. Z důvodu limitace prostorem stránky je zobrazeno pouze 8 hlavních komponent nového datasetu.

ip.src	ip.dst	_ws.col.Protocol
192.168.1.99	192.168.1.101	TCP
192.168.1.101	192.168.1.99	TCP
192.168.1.99	192.168.1.101	TCP
192.168.1.99	192.168.1.101	Modbus/TCP
192.168.1.99	192.168.1.103	TCP
192.168.1.103	192.168.1.99	TCP
192.168.1.101	192.168.1.99	Modbus/TCP
192.168.1.99	192.168.1.103	TCP
192.168.1.99	192.168.1.103	Modbus/TCP

Transformace do binární podoby

ip.src_nan	ip.src_192.168.1.99	ip.src_192.168.1.101	ip.src_192.168.1.103	ip.src_192.168.1.104	ip.src_192.168.1.105	ip.src_192.168.1.102	ip.src_192.168.1.106
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0

PCA transformace

0	1	2	3	4	5	6	7
0.397991	0.208493	0.002853	0.000737	0.337108	4,53E+10	-7,20E+10	-0.00042
0.397991	0.208493	0.002853	0.000737	0.337108	4,53E+10	-7,20E+10	-0.00042
-0.57985	0.222186	0.712971	-0.00229	-0.05401	-0.27334	-0.68462	0.535699
0.079893	-0.69903	-0.01049	0.710881	-0.09682	0.000963	6,92E+09	0.000189
-0.57985	0.222186	0.712971	-0.00229	-0.05401	-0.27334	-0.68462	0.535699
0.561763	0.712513	0.725538	-0.00017	-0.23184	-0.27388	-0.68482	0.53485
-0.58001	0.222416	0.712513	-0.00229	-0.05502	-0.34142	0.724352	0.434969
0.079768	-0.69922	-0.01049	0.710541	-0.09832	0.000959	1,57E+10	0.000194
0.078203	0.113279	0.002082	0.712994	-0.27465	0.000414	-0.00019	-0.00066
0.397991	0.208493	0.002853	0.000737	0.337108	4,53E+10	-7,20E+10	-0.00042

Obr. 40: Proces úpravy nominálních hodnot datasetu pro vybrané atributy (dataset 1). [vlastní zdroj]

Experimenty

První oblast experimentů je zaměřena na komparaci a výběr technik pro zpracování datasetů pro detekci anomálií. Zvláštní zřetel byl brán na techniky pro zpracování numerických dat. V rámci nominálních dat byl zvolen vhodný postup, respektující charakteristiky řešeného problému.

Pro tvorbu a evaluaci dílčích konfigurací technik byl zvolen dataset 1. Tento dataset je tvořen řadou pcap souborů, v nichž je uchována síťová komunikace ICS systému pro normální provoz a kybernetické útoky. V rámci pcap souborů bylo extrahováno 19 numerických atributů a 29 nominálních atributů jak pro trénovací dataset, tak pro testovací dataset. Trénovací dataset obsahuje 134 690 datových záznamů. Pro validační dataset bylo využito 30 % z trénovacího datasetu. Byly vybrány následující kybernetické útoky pro otestování prediktivního modelu ("CA1_1", "CA1_2", "CA1_3", CA1_4).

Pro experimenty byly vybrány následující algoritmy strojového učení. Jako první byla využita základní **neuronová síť** s architekturou autoenkodéru. Druhým algoritmem byla zvolena rekurentní neuronová síť **LSTM** s architekturou autoenkodéru. Třetím algoritmem strojového učení byl určen **Isolation Forest**. Poslední variantou byl zvolen **OCSVM** algoritmus s lineárním jádrem. Pro experiment bylo uvažováno se šesti technikami předzpracování dat. Tři techniky pro problematiku náhrady chybějících hodnot (aritmetický průměr, medián, konstanta - 0) a tři techniky pro změnu měřítka dat pomocí normalizace pro hodnoty (<-1,1>; <0,1>) a standardizace.

Výsledky byly získány prostřednictvím modelů, jejichž nastavení hyperparametrů je zvoleno jako základní. Tento postup byl zvolen z důvodu nalezení efektivního postupu úpravy dat pro účely detekce anomálií podle algoritmů strojového učení. Výsledky získané v rámci této sekce jsou využity pro výběr optimálního řešení úpravy datasetů pro další sekce v rámci této kapitoly. Pro evaluaci řešení byly použity následující metriky: M_{F1}, M_{MCC}, M_{Prec}, M_{FPR} a Čas.

Bylo vytvořeno 900 prediktivních modelů pro každý dílčí algoritmus strojového učení, prostřednictvím 9 trénovacích datasetů. Tyto datasety byly modifikovány podle vybraných technik pro úpravu numerických dat. Každý z uvedených datasetů představuje dílčí unikátní kombinaci těchto technik. Vytvořené modely byly použity k detekci anomálií v rámci čtyř testovacích datasetů. Každý z nich obsahuje jeden z vybraných kybernetických útoků.

Výsledky pro každý z algoritmů strojového učení jsou zobrazeny v přehledných tabulkách. Vzhledem k tomu, že první tři algoritmy (neuronová síť, LSTM, Isolation Forest) strojového učení jsou stochastické povahy bylo nutné výsledná řešení opakovat (100 opakování), aby získané výsledky měly statistickou významnost. Algoritmus OCSVM je deterministické povahy, tudíž

není potřeba žádný z experimentů opakovat. Pro porovnání všech variant zpracování dat v rámci každého algoritmu strojového učení bylo využito neparametrického testu – Friedmanův test [64]. V rámci tohoto testu je potvrzena nebo vyvrácena nulová hypotéza pomocí p-hodnoty. Pro zamítnutí nebo přijetí nulové hypotézy je uvažováno s hodnotou 5 %. Pro zhodnocení statistické významnosti výsledků Friedmanova testu je využito Nemenyiho testu pro definici kritické vzdálenosti. Ta určuje významné statistické rozdíly mezi daty.

V rámci dalšího řešení experimentů byl zvolen následující postup.

1. U každého algoritmu strojového učení je nejprve definováno nastavení daného algoritmu vycházející z jeho hyperparametrů.
2. Druhá část v rámci dílčích algoritmů je zaměřena na prezentaci výsledků ve formě tabulky pro metody zpracování dat v závislosti na vybraném kybernetickém útoku.
3. Poslední část je věnována definování pořadí pro jednotlivé techniky prostřednictvím Friedmanova testu. Techniky také jsou porovnány pomocí Nemenyiho kritické vzdálenosti. Ta je v grafech zobrazena jako černá čárkovaná čára.

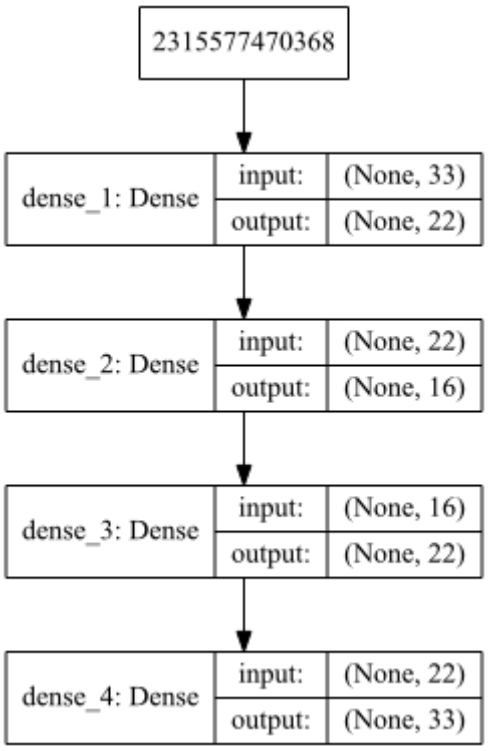
V poslední části jsou jednotlivé kombinace technik pro úpravu dat pro přehlednost značeny jako data1, data2 až data9. Toto značení odpovídá kombinaci technik. Například data1 odpovídají kombinaci (Aritmetický průměr; normalizace <0,1>) a data2 odpovídají kombinaci (Aritmetický průměr; normalizace <-1,1>). Následuje výčet algoritmů strojového učení spolu s jejich základním nastavením. Kompletní výsledky, včetně tabulek výsledků a grafů porovnání algoritmů pomocí Friedmanových testů jsou uvedeny v **Příloze A**.

Neuronová síť

V rámci zvoleného experimentu byla jako první využita neuronová síť, typu „backpropagation“ s architekturou symetrického autoenkodéru. Byla využita základní konfigurace se dvěma skrytými vrstvami podle diagramu uvedeného na Obr. 41. Pro vstup bylo využito 33 atributů, které korespondovaly s počtem hlavních komponent vytvořených prostřednictvím PCA algoritmu. Výčet nastavení hyperparametrů neuronové sítě je uveden v Tab. 9.

Tab. 9 – Hyperparametry využité pro nastavení neuronové sítě se strukturou autoenkodéru. [vlastní zdroj]

Hyperparametry	Hodnota
Skryté vrstvy	2
Umístění „bottle neck“	2
Optimazér	nadam
Chybová funkce	střední kvadratická chyba
Epochy	200
Velikost dávky	64
Aktivační funkce	elu



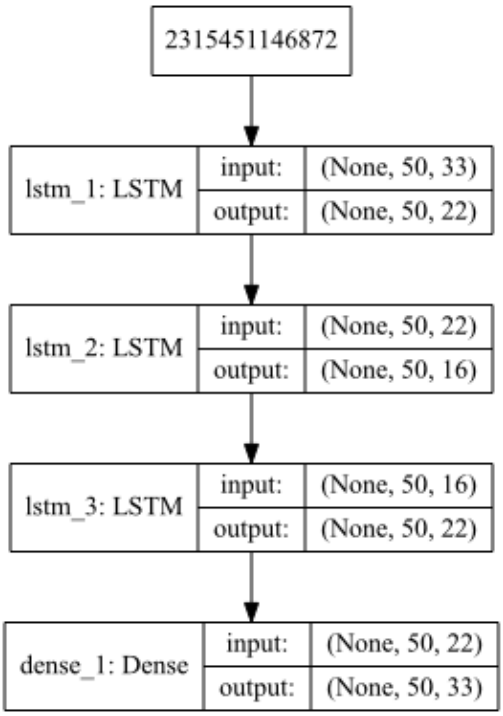
Obr. 41: Struktura autoenkodéru. [vlastní zdroj]

LSTM

V rámci zvoleného experimentu byla jako druhý algoritmus strojového učení využita rekurentní neuronová síť s architekturou symetrického autoenkodéru. Je využito základní konfigurace se dvěma skrytými vrstvami podle diagramu uvedeného na Obr. 42. Pro vstup bylo využito 33 atributů, které korespondují s počtem hlavních komponent vytvořených prostřednictvím PCA algoritmu. Výčet nastavení hyperparametrů LSTM je zobrazen v Tab. 10.

Tab. 10 – Hyperparametry využité pro nastavení neuronové sítě se strukturou autoenkodéru. [vlastní zdroj]

Hyperparametry	Hodnota
Skryté vrstvy	2
Umístění „bottle neck“	2
Optimazér	nadam
Chybová funkce	střední kvadratická chyba
Epochy	200
Velikost dávky	64
Aktivační funkce	elu
Dropout	0.1
Rekurentní dropout	0.1
Časový úsek	50



Obr. 42: Struktura autoenkodéru. [vlastní zdroj]

Isolation Forest

Jako třetí algoritmus strojového učení byl využit Isolation Forest (IF). Je využito základní konfigurace IF. Pro vstup bylo vybráno 33 atributů, které korespondují s počtem hlavních komponent vytvořených prostřednictvím PCA algoritmu. Výčet nastavení hyperparametrů IF je zobrazen v Tab. 11.

Tab. 11 – Hyperparametry využité pro nastavení IF. [vlastní zdroj]

Hyperparametry	Hodnota
Počet vzorků	256
Počet stromů	100
Kontaminace	0.1
Max počet atribut využitých pro rozdělení	1

OCSVM

V rámci zvoleného experimentu byl jako čtvrtý algoritmus strojového učení zvolen OCSVM algoritmus s radiálním jádrem. Pro vstup bylo využito 33 atributů, které korespondují s počtem hlavních komponent vytvořených prostřednictvím PCA algoritmu. Výčet nastavení hyperparametrů OCSVM je zobrazen v Tab. 12. Jedná se o deterministický algoritmus strojového učení. Z tohoto důvodu není potřeba opakování experimentů. Tento algoritmus má při stejných vstupech vždy identické výstupy.

Tab. 12 – Hyperparametry využité pro nastavení OCSVM. [vlastní zdroj]

Hyperparametry	Hodnota
Gamma	0.2

Pro každou kombinaci technik pro úpravu datasetu jsou vypočteny následující veličiny: maximální hodnota, minimální hodnota a průměr z vybraných hodnotících metrik. Výsledky pro algoritmu OCSVM jsou uvedeny v **Příloze A**. Tyto výsledky lze charakterizovat jako velmi slabé v rámci všech hodnotících kritérií. Zvláště pak závažné jsou výsledky v rámci metriky M_{FPR}. Tyto výsledky ve spojení s velmi vysokým časem pro detekci anomálií naznačuje zásadní nedostatky OCSVM pro detekci anomálií v prostředí ICS.

Diskuse dílčích výsledků

V rámci uskutečněného výzkumu popsaného v této kapitole byla provedena řada experimentů. Bylo využito čtyř algoritmů strojového učení pro detekci anomálií v rámci ICS systému (dataset 1). Tyto experimenty byly provedeny se záměrem pro definování vhodných úprav a zpracování numerických hodnot pro datasety. Byla provedena řada experimentů, jejichž výsledky jsou porovnány pomocí pořadí v závislosti na Friedmanových testech a Nemenyiho kritické vzdálenosti.

Z výsledků pro neuronovou síť lze vyvodit následující závislosti. Techniky pro úpravu numerických dat se v daných případech (náhrada konstantou; normalizace <-1,1>), (aritmetický průměr; standardizace), (medián; standardizace), (náhrada konstantou; standardizace) zásadně liší od zbylých technik. Lze tedy konstatovat nevhodnost těchto technik pro detekci anomálií v případě využití neuronových

sítí. Z provedených testů lze identifikovat kombinaci technik, které poskytují nejlepší výsledky ve většině případů kybernetických útoků. Z tohoto důvodu bude nadále využíváno v dalších experimentech aritmetického průměru pro náhradu prázdných hodnot a normalizaci dat v rozmezí <0,1> pro neuronové sítě s architekturou autoenkodéru.

Výsledky pro rekurentní neuronovou síť LSTM vypovídají o nevhodnosti použití následujících technik pro úpravu numerických dat. Výsledky technik (aritmetický průměr; standardizace), (medián; normalizace <-1,1>), (medián; standardizace), (náhrada konstantou; normalizace <-1,1>), (náhrada konstantou; standardizace) jsou podstatně horší ve spojení s detekcí anomálií pro algoritmus LSTM. Z provedených testů lze identifikovat kombinaci technik, která poskytuje nejlepší výsledky ve všech případech kybernetických útoků. Proto bude nadále využíváno v dalších experimentech aritmetického průměru pro náhradu prázdných hodnot a normalizaci dat v rozmezí <0,1> pro rekurentní neuronové sítě LSTM s architekturou autoenkodéru.

Výsledky pro algoritmus strojového učení Isolation Forest jsou využity pro identifikaci nevhodných technik pro úpravu numerických dat, mezi které se řadí kombinace technik: (aritmetický průměr; normalizace <0,1>), (náhrada konstantou; normalizace <0,1>). Jako nejlepší varianty byly identifikovány kombinace technik (aritmetický průměr; standardizace) a (náhrada konstantou; standardizace**).** Zajímavou skutečností v případě IF je poměrně vysoké umístění technik, jejichž součástí je využití standardizace pro změnu formátu datasetu. Oproti neuronovým sítím, kde standardizace využitá pro úpravu datasetu často vychází jako nejhorší možnost. Určující kombinace technik pro zpracování numerických dat, která bude využita pro následující experimenty, byla zvolena (aritmetický průměr; standardizace). Tato zvolená kombinace technik pro úpravu datasetu vykazuje velmi dobré hodnoty pro značně důležité metriky, jakými jsou M_{FPR} a čas.

Souhrnné výsledky jsou uvedeny v tabulce Tab. 13, kde jsou porovnány. Tato tabulka vychází z provedených Friedmanových testů (viz. **Příloha A**). Jednotlivé příklady ohodnoceny od nejlepšího (nejmenší hodnota) po nejhorší (nejvyšší hodnota).

Tab. 13 – Souhrnné výsledky experimentu – úprava datasetu. [vlastní zdroj]

Kombinace technik/ algoritmy	data 1	data 2	data 3	data 4	data 5	data 6	data 7	data 8	data 9
Neuronová síť	5	6	1	4	3	2	4	8	7
	1	5	8	2	4	7	3	3	6
	1	2	6	3	5	8	4	4	7
	1	5	7	3	2	8	4	5	6
Suma	8	18	22	12	14	25	15	20	26
LSTM	1	4	7	2	5	6	3	9	8
	1	4	7	2	5	8	3	6	9
	1	6	9	2	4	8	3	5	7
	1	4	9	2	6	8	3	5	7
Suma	4	18	32	8	20	30	12	25	31
IF	8	3	2	7	5	6	9	4	1
	8	4	1	7	6	2	9	5	3
	4	8	3	2	9	5	6	7	1
	6	4	1	5	7	2	8	9	3
Suma	26	19	7	21	27	15	32	25	8

Posledním algoritmem strojového učení, který byl využit pro detekci anomálií, v případě ICS systémů byl zvolen OCSVM. Jedná se o deterministický algoritmus, který podle výsledků vykazuje velmi slabé až nedostačující charakteristiky pro řešenou problematiku pro všechny z kombinací technik pro úpravu numerických dat. Jednotlivé výsledky jsou prakticky identické. Při srovnání výsledků v podobě jednotlivých metrik byla pro další experimenty vybrána následující kombinace technik pro úpravu numerických dat (Aritmetický průměr; normalizace <-1,1>).

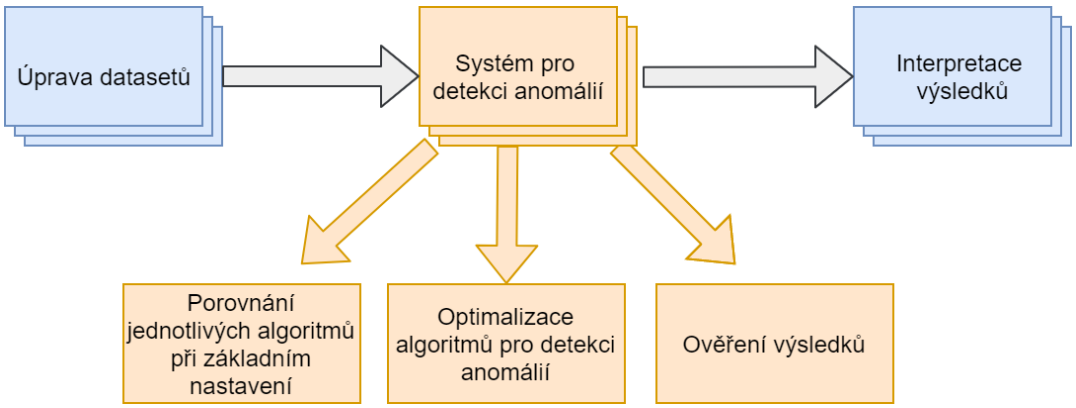
V této kapitole byla provedena řada experimentů, jejichž účelem bylo definovat optimální kombinaci technik pro úpravu datasetu. Výsledky poukazují na rozdílnost optimální kombinace technik pro různé algoritmy strojového učení. Provedený výzkum poukazuje na nutnost rozdílného přístupu k úpravě dat pro jednotlivé algoritmy strojového učení, a také nevhodnost implementace jednoho řešení pro všechny algoritmy.

6.2.2 Postup nastavení a ohodnocení jednotlivých algoritmů strojového učení pomocí optimalizačních technik

Tato podkapitola disertační práce je věnována definování postupu výběru nejlepšího algoritmu strojového učení pro detekci anomálií v rámci systémů ICS. K dosažení tohoto cíle jsou využity optimalizační algoritmy pro určení hyperparametrů algoritmů strojového učení. Cílem tohoto výzkumu je nalezení

nejvhodnějšího algoritmu pro detekci anomálií podle zvolených metrik. Druhým výsledkem této kapitoly je definování postupu pro vytvoření systému pro detekci anomálií, který je do určité míry variabilní. Toho lze dosáhnout prostřednictvím optimalizačních algoritmů, které dokáží upravit zvolený detekční algoritmus tak, aby respektoval charakteristiky systému, ve kterém bude popisovaný systém nasazený.

Tato kapitola je rozdělena na tři základní podkapitoly viz. Obr. 43. První podkapitola je zaměřena na porovnání jednotlivých algoritmů strojového učení při jejich neměnném nastavení. Tento experiment je proveden pro rozdílné datasety. Druhá podkapitola je zaměřena na zhodnocení detekčních vlastností zvolených algoritmů při proměnném nastavení definovaným prostřednictvím optimalizačních technik. Tyto algoritmy jsou testovány na rozdílných datasetech stejně jako v případě první podkapitoly. Na výsledky těchto podkapitol navazuje třetí podkapitola, kde je provedeno ověření výsledků. V ní jsou prověřeny získaná nastavení algoritmů strojového učení na kybernetických útocích. Tyto kybernetické útoky nebyly využity v předešlých kapitolách.



Obr. 43: Diagram procesů pro tvorbu algoritmu – optimalizace. [vlastní zdroj]

Porovnání algoritmu strojového učení při základním nastavení.

Tato podkapitola je zaměřena na porovnání algoritmů strojového učení využitých pro detekci anomálií. Je využito statického nastavení algoritmů. Toto nastavení bylo popsáno v předešlé kapitole 5.2.1. Zmíněné nastavení se týká jednak úpravy numerických hodnot v rámci trénovacího, validačního a testovacího datasetu, ale i nastavení hyperparametrů v případě jednotlivých algoritmů. Porovnány jsou jednotlivé algoritmy mezi sebou v rámci rozdílných datasetů. Hlavním cílem této podkapitoly je základní zhodnocení vybraných algoritmů strojového učení pro standartní nastavení hyperarametrů algoritmů strojového učení v rámci proměnného prostředí prostřednictvím rozdílných datasetů. Získané výsledky budou využity při komparaci výsledků v rámci druhé podkapitoly. Tedy s výsledky získanými při využití optimalizačních algoritmů. Závěr této kapitoly je zaměřen na ověření vytvořeného řešení (systému detekce

anomálií) pomocí kybernetických útoků, které nebyly využity pro tvorbu tohoto systému.

Pro nalezení vítězného řešení mezi stochastickými algoritmy je využito Friedmanova testu pro získání pořadí jednotlivých řešení. Techniky jsou také porovnány pomocí Nemenyiho kritické vzdálenosti. Výsledky získané prostřednictvím dílčích algoritmů strojového učení jsou shrnuty v **Příloze B**. Pro stochastické algoritmy: neuronová síť, LSTM a Isolation forest jsou uvedeny maximální, minimální a průměrné hodnoty výsledků metrik jednotlivých algoritmů. V rámci každého stochastického algoritmu bylo vytvořeno 100 prediktivních modelů z důvodu jejich testování. Pro deterministický algoritmus OCSVM je uvedena jedna výsledná hodnota.

Porovnání algoritmu strojového učení pro dataset 1

Tato sekce slouží ke zhodnocení zvolených algoritmů pro detekci anomálií v rámci datasetu 1. Tento dataset byl publikován v rámci publikace. [43] V rámci zvoleného ICS systému jsou využity čtyři testovací datasety. Každý odpovídá jednomu ze zvolených kybernetických útoků (CA1_1 až CA1_4). Každý ze zvolených testovacích datasetů je využit pro zhodnocení detekčních schopností v rámci vybraných detekčních algoritmů. Podrobné výsledky metrik pro jednotlivé algoritmy strojového učení při základním nastavením jsou v **Příloze B** včetně grafů srovnávající algoritmy pomocí Friedmanových testů.

Porovnání algoritmu strojového učení pro dataset 2

V této sekci jsou publikovány výsledky získané při detekci kybernetických útoků pro dataset 2 [44]. Bylo využito algoritmů strojového učení se základním nastavení hyperparametrů. V rámci zvoleného ICS systému je využito šesti testovacích datasetů, přičemž každý odpovídá jednomu ze zvolených kybernetických útoků (CA2_1 až CA2_6). Každý ze zvolených testovacích datasetů je využit pro zhodnocení detekčních schopností v rámci vybraných detekčních algoritmů. Podrobné výsledky metrik pro jednotlivé algoritmy strojového učení při základním nastavením jsou v **Příloze B** včetně grafů srovnávajících algoritmy pomocí Friedmanových testů.

Porovnání algoritmu strojového učení pro dataset 3

V rámci této sekce jsou publikovány výsledky získané při detekci kybernetických útoků pro dataset 3. [45] Bylo využito algoritmů strojového učení se základním nastavením hyperparametrů. Pro zvolený ICS systém je využito šesti testovacích datasetů. Každý odpovídá jednomu ze zvolených kybernetických útoků (CA3_1 až CA3_6). Všechny zvolené testovací datasety jsou využity pro zhodnocení detekčních schopností v rámci vybraných detekčních algoritmů. Podrobné výsledky metrik pro jednotlivé algoritmy strojového učení při základním nastavením jsou v **Příloze B** včetně grafů srovnávajících algoritmy pomocí Friedmanových testů.

Diskuse dílčích výsledků

V této podkapitole byly prezentovány výsledky pro jednotlivé algoritmy strojového učení se základním nastavením hyperparametrů. V rámci experimentů bylo využito tří datasetů včetně kybernetických útoků. Z výsledků lze dedukovat poměrně malou variabilitu mezi využitými algoritmy (především pro dataset 3). V dané fázi výzkumu nelze detekovat významnější rozdíl mezi základní neuronovou sítí a rekurentní sítí LSTM. V rámci algoritmu Isolation forest je situace však odlišná. V řadě případů lze identifikovat výsledky tohoto algoritmu jako výrazně horší než u neuronové sítě, popřípadě LSTM. Poslední deterministický algoritmus OCSVM vykazuje horší výsledky než předešlé popsané algoritmy, a to především v nejdůležitějších metrikách hodnocení, tedy množství falešně identifikovaných kybernetických útoků (měřeno pomocí M_{FPR}). Druhou metrikou je čas potřebný ke klasifikaci jednotlivých záznamů. Souhrnné výsledky jsou představeny v Tab. 14. Zde je každý algoritmus ohodnocen pomocí Friedmanova testu. V této tabulce je prezentováno pořadí mezi jednotlivými algoritmy strojového učení při základním nastavení. Menší hodnota pořadí značí lepší detekční vlastnosti.

Tab. 14 – Souhrnné výsledky experimentu – porovnání algoritmů strojového učení při základním nastavení. [vlastní zdroj]

Zastoupené algoritmy/ datasety	**Neuronová síť**	**LSTM**	**Isolation forest**
Dataset 1	2	1	3
	2	1	3
	2	1	3
	2	1	3
Dataset 2	1	2	1
	1	2	3
	3	2	1
	1	2	3
	3	2	1
	1	2	3
Dataset 3	1	3	2
	2	3	1
	3	2	1
	1	2	3
	1	2	3
	3	2	1
Suma	29	30	35

Vzhledem k horším detekčním vlastnostem algoritmu OCSVM v prakticky u všech provedených měření byl proveden experiment. Na jeho základě bylo třeba rozhodnout, jestli je příhodné se nadále se zmíněným algoritmem zabývat. Byly prověřeny detekční schopnosti OCSVM pro různé hodnoty gamma

hyperparametru. Gamma je v podstatě jediný hyperparametr, který je nutno nastavit v rámci OCSVM. Z výsledků v **Příloze C** lze konstatovat horší vlastnosti OCSVM zvláště u nejvýznamnějších metrik (M_{FPR} a času) v prakticky všech případech. Na základě těchto výsledků byl algoritmus OCSVM vyloučen z dalších experimentů.

Využití optimalizačních technik pro nastavení algoritmů strojového učení

Tato podkapitola disertační práce je zaměřena na popsání procesu a diskusi výsledků optimalizace hyperparametrů pro vybrané algoritmy strojového učení. Byly vybrány tři algoritmy pro optimalizaci: evoluční algoritmus, random choise a TPE. Tyto optimalizační techniky byly využity pro algoritmy strojového učení LSTM, neuronová síť a IF v rámci vybraných datasetů. Každý z algoritmů strojového učení využívá řadu hyperparametrů, které více či méně ovlivňují jejich funkci. Optimálně nastavený algoritmus strojového učení se vyznačuje efektivním chodem, a proto je vhodné využít optimalizace i pro detekci anomálií.

Každá z následujících sekcí této podkapitoly je věnována jednomu z optimalizačních algoritmů. V každé z těchto sekcí jsou prezentovány výsledky pro jednotlivé algoritmy strojového učení prostřednictvím vybraných datasetů. Výsledná nastavení algoritmů získaná pomocí optimalizačních algoritmů jsou následně evaluována z důvodu ověření vhodnosti získaného řešení. Popis využitých hodnot pro jednotlivé hyperparametry pro každý z algoritmů strojového učení je uveden v následující kapitole Genetický algoritmus. Tyto hyperparametry jsou využity pro další optimalizační algoritmy Random search a TPE.

Poměrně často bylo v této kapitole využito pojmu **zástupce**. V kontextu této kapitoly je toto názvosloví využito pro popis jednoho algoritmu strojového učení včetně jeho nastavení pomocí hyperparametrů.

V rámci řešeného problému je nutné vyřešit problematiku vícekriteriální optimalizace. Z tohoto důvodu je využit algortitmus multikriteriálního hodnocení TOPSIS, pro výpočet tzv „objective function“. Tato funkce je základem pro provedení optimalizace zvoleného řešení. Pro TOPSIS byly využity následující metriky: **M_{F1}, M_{MCC}, M_{Prec}, M_{FPR}, Čas**. Pro tyto metriky byl vytvořen Fullerův trojúhelník, kde pomocí párového srovnání byly nalezeny závislosti mezi jednotlivými metrikami viz. Tab. 15

Tab. 15 – Výsledný Fullerův trojúhelník pro vybrané metriky. [vlastní zdroj]

Čas	Čas	Čas	Čas
M_{FPR}	M_{F1}	M_{MCC}	M_{PREC}
M_{FPR}	M_{FPR}	M_{FPR}	
M_{F1}	M_{MCC}	M_{PREC}	
M_{F1}	M_{F1}		
M_{MCC}	M_{PREC}		
M_{MCC}			
M_{PREC}			

V rámci Fullerova trojúhelníku jsou postupně srovnávány jednotlivé metriky z pohledu jejich významnosti. Oranžová barva v rámci popisované tabulky demonstruje vyšší významnost kritéria oproti kritériu, se kterým je porovnáváno. V návaznosti na Tab. 15 byly vypočítány váhy $W_{i,j}$. Tyto váhy představují míru významnosti jednotlivých kritérii. Výsledné hodnoty kritérií jsou znázorněny v Tab. 16.

Tab. 16 – Výsledné váhy pro jednotlivé metriky. [vlastní zdroj]

Metriky/ váhy	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**Čas**	**M_{FPR}**
Váhy	0.12	0.16	0.17	0.2	0.35

Genetický algoritmus

Tato sekce byla zaměřena na popis postupu a výsledků spojených s aplikací genetického algoritmu v prostředí algoritmů strojového učení. Genetický algoritmus byl využit k optimalizaci hyperparametrů jednotlivých algoritmů strojového učení. U každého GA je nastavena populace o velikosti 80 jedinců. Tato populace byla křížena po dobu 80 generací. Pro každý algoritmus strojového učení a dataset byl tento postup opakován desetkrát z důvodu ověření výsledků této nedeterministické úlohy. Na konci každé generace bylo zachováno 40 % nejlepších jedinců. Zbylých 60 % populace bylo nahrazeno novými jedinci, kteří vznikají kombinací nejlepších jedinců. Mutace v rámci jedné generace byla nastavena na hodnotu 6 % a náhodný výběr při selekci jedinec je nastaven na 4 %.

Pro výběr výsledného nastavení hyperparametrů byl zvolen následující postup. U každého z deseti průběhů genetického algoritmu bylo zvoleno 20 % nejlepších jedinců z poslední generace podle filozofie založené na Paretově pravidle. V každé z vytvořených množin řešení bylo vybráno jedno finální řešení, které bylo nejvíce zastoupeno. Výsledkem bylo deset řešení, reprezentujících každého

z deseti průběhů genetického algoritmu. Tento výběr byl dále zredukován podle stejného postupu na jedno řešení. Výsledné řešení bylo využito jako podklad pro další experimenty. V každém z nich bylo vytvořeno výsledné nastavení hyperparametrů pro 100 nových modelů, které byly otestovány. Tento postup byl opakován pro každý z genetických algoritmu, pro každý z vybraných datasetů. Výsledky byly poté porovnány se základním nastavením hyperparametrů algoritmů strojového učení, které bylo uskutečněno v předcházející kapitole.

Provedené experimenty v rámci kapitoly poukazují na řadu kombinací hyperparametrů, které se blíží optimálnímu nastavení algoritmu ANN pro detekci anomálií. Finální kombinace hyperparametrů byla vybrána podle následujícího postupu. Z poslední generace bylo vybráno 20 nejlepších jedinců podle skóre v rámci každého z průběhů genetických algoritmů. Z těchto hodnot je pro každý průběh genetického algoritmu vybrána kombinace hyperparametrů, která je nejčastěji zastoupena. Výsledná kombinace byla poté získána stejným postupem porovnání všech „vítězných“ kombinací pro všechny průběhy genetického algoritmu (10 průběhů).

V následujících odstavcích je popsáno nastavení GA pro každý dílčí algoritmus strojového učení, včetně výsledků procesu optimalizace.

Neuronová síť

Pro algoritmus strojového učení ANN jsou definovány následující hodnoty hyperparametrů (viz. Tab. 17), které jsou využity v rámci genetického algoritmu pro vyhledání optimálního nastavení algoritmu. Pro jednotlivé datasety jsou zvoleny rozdílné maximální počty atributů z důvodu rozdílné dimenze každého z datasetů.

Tab. 17 Zvolené hyperparametry pro genetický algoritmus v rámci neuronové sítě. [vlastní zdroj]

Hyperparametry	Hodnoty hyperparametrů
Počet neuronů	Dataset 1{4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43} Dataset 2{4, 5, 6, 7, 8, 9, 10} Dataset 3{4, 5, 6, 7, 8, 9, 10}
Počet vrstev	{5, 7, 9, 11, 13, 15}
Velikost dávky	{64, 128, 256}
Počet epoch	{100, 150, 200, 250, 300}
Velikost "dropout"	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Aktivační funkce	{relu, elu, tanh, sigmoid}
Optimalizační algoritmus	{rmsprop, adam, adagrad, adamax, nadam}
Míra učení	{0.00003, 0.00005, 0.00008, 0.0001, 0.0005, 0.001, 0.002, 0.005, 0.008}
Počet neuronů pro "bottleneck" 1	"bottleneck" 1<počet neuronů v síti
Počet neuronů pro "bottleneck" 2	"bottleneck" 1≤"bottleneck" 2≤ počet neuronů v síti

Prvním diskutovaným algoritmem strojového učení je neuronová síť. V této sekce je představena finální kombinace hyperparametrů, která bude využita pro otestování získaného řešení pro každý dataset. V tabulce (viz. Tab. 18) jsou prezentovány vybrané kombinace hyperparametrů pro dílčí datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.7. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.77 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.8. Výsledky optimalizace jsou k dispozici v **Příloze D**.

Tab. 18 Zvolené hyperparametry pro neuronovou síť optimalizovanou pomocí genetického algoritmu pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	**Dataset 1**	**Dataset 2**	**Dataset 3**
Počet neuronů	37	4	8
Počet vrstev	5	5	11
Velikost dávky	128	64	64
Počet epoch	150	100	100
Velikost “dropout”	0.1	0.5	0.4
Aktivační funkce	elu	tanh	elu
Optimalizační algoritmus	rmsprop	nadam	nadam
Míra učení	0.0005	8e-05	0.008
Počet neuronů pro “bottleneck” 1	22	3	3
Počet neuronů pro “bottleneck” 2	33	3	5

LSTM

Algoritmus LSTM byl vybrán jako druhý zástupce algoritmů strojového učení pro definované experimenty. Jako u předcházejícího zástupce jsou i zde definovány následující hodnoty hyperparametrů (viz. Tab. 19), které jsou využity v rámci genetického algoritmu pro vyhledání optimálního nastavení algoritmu. Pro jednotlivé datasety jsou zvoleny rozdílné maximální počty atributů z důvodu rozdílné dimenze každého z datasetů.

Tab. 19 Zvolené hyperparametry pro genetický algoritmus v rámci LSTM. [vlastní zdroj]

Hyperparametry	Hodnoty hyperparametrů
Počet neuronů	Dataset 1{4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43} Dataset 2{4, 5, 6, 7, 8, 9, 10} Dataset 3{4, 5, 6, 7, 8, 9, 10}
Počet vrstev	{5, 7, 9, 11, 13, 15}
Velikost dávky	{64, 128, 256}
Počet epoch	{100, 150, 200, 250, 300}
Velikost rekurentního "dropout"	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Velikost "dropout"	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Aktivační funkce	{relu, elu, tanh, sigmoid}
Optimalizační algoritmus	{rmsprop, adam, adagrad, adamax, nadam}
Míra učení	{0.00003, 0.00005, 0.00008, 0.0001, 0.0005, 0.001, 0.002, 0.005, 0.008}
Počet neuronů pro "bottleneck" 1	"bottleneck" 1<počet neuronů v síti
Počet neuronů pro "bottleneck" 2	"bottleneck" 1≤"bottleneck" 2≤ počet neuronů v síti

Druhým diskutovaným algoritmem strojového učení je algoritmus LSTM. V této sekci je představena finální kombinace hyperparametrů, která bude využita pro otestování získaného řešení pro každý dataset. V tabulce (viz. Tab. 20) jsou prezentovány vybrané kombinace hyperparametrů pro dílčí datasety u probíraného algoritmu LSTM. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.68. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.67 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.8. Výsledky optimalizace jsou k dispozici v **Příloze D**.

Tab. 20 Zvolené hyperparametry pro LSTM optimalizovanou pomocí genetického algoritmu pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	**Dataset 1**	**Dataset 2**	**Dataset 3**
Počet neuronů	37	4	4
Počet vrstev	5	5	9
Velikost dávky	256	64	64
Počet epoch	300	100	200
Velikost “dropout”	0.2	0.3	0.3
Velikost rekurentního “dropout”	0.2	0.3	0.4
Aktivační funkce	elu	tanh	elu
Optimalizační algoritmus	adamax	rmsprop	nadam
Míra učení	0.008	0.001	0.001
Počet neuronů pro “bottleneck” 1	34	3	1
Počet neuronů pro “bottleneck” 2	35	3	3

Isolation forest

Pro algoritmus strojového učení IF jsou definovány následující hodnoty hyperparametrů (viz. Tab. 21), které jsou využity v rámci genetického algoritmu pro vyhledání optimálního nastavení algoritmu. U jednotlivých datasetů jsou zvoleny rozdílné maximální počty atributů z důvodu rozdílné dimenze každého z datasetů.

Tab. 21 Zvolené hyperparametry pro genetický algoritmus v rámci IF. [vlastní zdroj]

Hyperparametry	Hodnoty hyperparametrů
Maximální počet atributů	Dataset 1 {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} Dataset 2{1,2,3,4,5,6,7,8,9,10} Dataset 3{1,2,3,4,5,6,7,8,9,10}
Počet vzorků	{50,75,100,125,150,175,200,225,250,275,300,325,350,375,400}
Počet stromů	{50,60,75,85,100,110,125,135,150,160,175,185,200,210,225,235, 250,260,275,285,300}
Kontaminace	{0,0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5}

Posledním diskutovaným algoritmem strojového učení je algoritmus IF. V rámci této sekce je představena finální kombinace hyperparametrů, která bude využita pro otestování získaného řešení pro každý dataset. V tabulce (Tab. 22) jsou prezentovány vybrané kombinace hyperparametrů pro dílčí datasety probíraného algoritmu IF. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.68. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.68 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.72. Výsledky optimalizace jsou k dispozici v **Příloze D**.

Tab. 22 Zvolené hyperparametry pro IF optimalizovanou pomocí genetického algoritmu pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	Dataset 1	Dataset 2	Dataset 3
Maximální počet atributů	1	1	7
Počet vzorků	400	50	375
Počet stromů	300	50	300
Kontaminace	0.05	0	0

Random Search

Tato sekce je zaměřena na popis postupu a výsledků spojených s aplikací optimalizačního algoritmu Random Search (RS) ve spojení s algoritmy strojového učení. RS je využit k optimalizaci hyperparametrů jednotlivých

algoritmů strojového učení. Množina hyperparametrů je pro každý algoritmus strojového učení definována v předchozí kapitole (Genetický algoritmus).

Jelikož RS nevyužívá žádný iterativní proces pro zlepšování výsledků byl využit jiný postup vzhledem ke GA. Jednotliví zástupci jsou vybíráni náhodně (náhodný výběr hyperparametrů), přičemž jednotlivé výsledky nejsou využity pro další optimalizaci. Z tohoto důvodu byl pro každý průběh optimalizačního algoritmu vybrán jedinec s nejvyšším skóre. Finální výběr nejvhodnějšího jedince byl proveden porovnáním nejlepších jedinců z deseti průběhů optimalizačního algoritmu. Finální jedinec představuje nejčastější kombinaci hyperparametrů z deseti průběhů, popřípadě pokud to není možné, tak jedince s nejvyšším skóre. V rámci následujících odstavců je popsáno nastavení jednotlivých algoritmů strojového učení pomocí optimalizačního algoritmu RS.

Neuronová síť

Prvním diskutovaný algoritmem strojového učení je neuronová síť. V této sekci je představena finální kombinace hyperparametrů, která bude využita pro otestování získaného řešení pro každý dataset. V tabulce (Tab. 23) jsou prezentovány vybrané kombinace hyperparametrů pro dílčí datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.71. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.73 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.79.

Tab. 23 Zvolené hyperparametry pro neuronovou síť optimalizovanou pomocí RS pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	**Dataset 1**	**Dataset 2**	**Dataset 3**
Počet neuronů	31	7	9
Počet vrstev	5	9	11
Velikost dávky	64	64	64
Počet epoch	200	100	250
Velikost “dropout”	0.2	0.3	0.6
Aktivační funkce	elu	elu	elu
Optimalizační algoritmus	rmsprop	rmsprop	adam
Míra učení	0.0001	0.001	0.005
Počet neuronů pro “bottleneck” 1	28	1	4
Počet neuronů pro “bottleneck” 2	28	5	7

LSTM

Druhým diskutovaným algoritmem strojového učení je LSTM. V této sekci je představena finální kombinace hyperparametrů LSTM pro dílčí datasety. V tabulce Tab. 24 jsou prezentovány vybrané kombinace hyperparametrů pro jednotlivé datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.67. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.67 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.78.

Tab. 24 Zvolené hyperparametry pro LSTM optimalizovanou pomocí RS pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	**Dataset 1**	**Dataset 2**	**Dataset 3**
Počet neuronů	19	8	5
Počet vrstev	5	5	5
Velikost dávky	128	256	128
Počet epoch	150	200	300
Velikost “dropout”	0.2	0.6	0.5
Velikost rekurentního “dropout”	0.1	0.6	0.2
Aktivační funkce	elu	elu	sigmoid
Optimalizační algoritmus	rmsprop	rmsprop	adamax
Míra učení	0.005	8e-05	5e-05
Počet neuronů pro “bottleneck” 1	12	6	4
Počet neuronů pro “bottleneck” 2	17	7	4

Isolation forest

Třetím diskutovaným algoritmem strojového učení je IF. V této sekci je představena finální kombinace hyperparametrů IF pro dílčí datasety. V tabulce Tab. 25 jsou prezentovány vybrané kombinace hyperparametrů pro jednotlivé datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.68. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.7 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.75.

Tab. 25 Zvolené hyperparametry pro IF optimalizovanou pomocí RS pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	Dataset 1	Dataset 2	Dataset 3
Maximální počet atributů	1	5	10
Počet vzorků	75	100	125
Počet stromů	250	75	225
Kontaminace	0.05	0.1	0.2

Tree-structured Parzen Estimator

Poslední část této kapitoly je zaměřena na prezentaci výsledků získaných prostřednictvím optimalizace algoritmů strojového učení: neuronová síť, LSTM a IF pomocí optimalizačního algoritmu TPE. Stejně jako v předešlých případech jsou jednotlivé algoritmy probírány v definovaném pořadí (neuronová síť, LSTM, IF). Každý z těchto algoritmů byl spuštěn paralelně desetkrát po dobu maximálně 10 000 iterací. Následuje popis nastavení algoritmů strojového učení, které vychází z provedených experimentů v rámci metacentra.

Neuronová síť

Jako první popisovaný algoritmus strojového učení je neuronová síť. Podobně jako v předcházejícím případu optimalizačního algoritmu RS jsou i zde popsány vybrané hyperparametry neuronové sítě. V této sekci je představena finální kombinace hyperparametrů, která je využita pro otestování získaného řešení pro každý dataset. V tabulce (Tab. 26) jsou prezentovány vybrané kombinace hyperparametrů pro dílčí datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.72. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.77 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.8.

Tab. 26 Zvolené hyperparametry pro neuronovou síť optimalizovanou pomocí TPE pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	**Dataset 1**	**Dataset 2**	**Dataset 3**
Počet neuronů	40	7	9
Počet vrstev	5	5	9
Velikost dávky	64	64	64
Počet epoch	200	200	250
Velikost “dropout”	0.1	0.9	0.5
Aktivační funkce	elu	tanh	elu
Optimalizační algoritmus	nadam	adagrad	adam
Míra učení	0. 001	0.0001	0.008
Počet neuronů pro “bottleneck” 1	25	5	4
Počet neuronů pro “bottleneck” 2	39	5	8

LSTM

Druhým diskutovaným algoritmem strojového učení je LSTM. V této sekci je představena finální kombinace hyperparametrů LSTM pro dílčí datasety. V tabulce Tab. 27 jsou prezentovány vybrané kombinace hyperparametrů pro jednotlivé datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.7. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.7 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.79.

Tab. 27 Zvolené hyperparametry pro LSTM optimalizovanou pomocí TPE pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	**Dataset 1**	**Dataset 2**	**Dataset 3**
Počet neuronů	43	8	10
Počet vrstev	7	9	7
Velikost dávky	64	64	128
Počet epoch	300	250	250
Velikost “dropout”	0.3	0.4	0.3
Velikost rekurentního “dropout”	0.1	0.5	0.6
Aktivační funkce	elu	tanh	tanh
Optimalizační algoritmus	rmsprop	rmsprop	nadam
Míra učení	0.001	0.001	0.001
Počet neuronů pro “bottleneck” 1	25	1	8
Počet neuronů pro “bottleneck” 2	37	4	8

Isolation forest

Třetí diskutovaný algoritmus strojového učení je IF. V této sekci je představena finální kombinace hyperparametrů IF pro dílčí datasety. V tabulce Tab. 28 jsou prezentovány vybrané kombinace hyperparametrů pro jednotlivé datasety v probírané neuronové síti. Pro dataset 1 byla vybrána varianta s maximálním skóre 0.69. Pro dataset 2 byla vybrána varianta s maximálním skóre 0.74 a pro dataset 3 byla vybrána varianta s maximálním skóre 0.75.

Tab. 28 Zvolené hyperparametry pro IF optimalizovanou pomocí TPE pro jednotlivé datasety. [vlastní zdroj]

Datasety/vybrané hyperparametry	Dataset 1	Dataset 2	Dataset 3
Maximální počet atributů	1	6	5
Počet vzorků	375	50	375
Počet stromů	200	235	250
Kontaminace	0.05	0.1	0

Diskuse dílčích výsledků

Tato podkapitola je zaměřena na diskusi výsledků nastavení hyperparametrů pomocí optimalizačních algoritmů. V této podkapitole byla provedena optimalizace algoritmu strojového učení v oblasti detekce anomálií vztahující se ke kybernetické bezpečnosti. V rámci výzkumu bylo vytvořeno velké množství experimentů, které reflektovaly dílčí kombinace mezi algoritmy strojového učení (neuronová síť, LSTM, IF) na jedné straně a optimalizačními algoritmy (evoluční algoritmus, RS, TPE) na straně druhé. Každá z prezentovaných kombinací byla provedena pro každý ze tří popisovaných datasetů. Jednotlivé experimenty byly poté desetkrát spuštěny v rámci metacentra z důvodu stochastického charakteru vybraných algoritmů. Toto metacentrum poskytlo nezbytnou výpočetní kapacitu pro provedení těchto úloh. V celkovém počtu bylo provedeno 270 dílčích experimentů, přičemž každý z experimentů probíhal maximálně 300 hodin. Celkově se tedy jedná o 81 000 hodin experimentů. Takto rozsáhlou výpočetní úlohu nebylo bez outsourcovaných zdrojů reálné možné vyřešit v reálném čase. Z tohoto důvodu bylo nutné využito služeb metacentra.

Finální zástupce z každé navržené kombinace byl vybrán v závislosti na skóre. Toto skóre představuje výsledek multikriteriálního hodnocení. Váhy multikriteriálního hodnocení jsou nastaveny v závislosti na specifikách, charakteru a potřebách ICS systémů. Samotné skóre je detailně popsáno v kapitole („Využití optimalizačních technik pro nastavení algoritmů strojového učení“). Z prezentovaných výsledků lze vyvodit následující závěry:

1. V závislosti na skóre vykazuje nejlepší výsledky neuronová síť oproti ostatním algoritmům strojového učení prakticky ve všech případech (datasetech, optimalizačních algoritmech).
2. Téměř stejné výsledky byly zaznamenány i v případě neuronové sítě optimalizované pomocí evolučního algoritmu a TPE. Tento trend lze pozorovat i v případě algoritmu LSTM a IF.

3. Z výsledků vyplývá „dominance“ dvou optimalizačních algoritmů v rámci provedených experimentů (evoluční algoritmus a TPE), na které je vhodné se zaměřit v následujících experimentech.

Zhodnocení výsledků algoritmů strojového učení pro jednotlivá nastavení hyperparametrů

Vybrané hyperparametry pro jednotlivé algoritmy z předešlé kapitoly je nutné otestovat v rámci zvolených datasetů, tak aby byla nalezena nejlepší varianta nastavení hyperparametrů pro zvolenou úlohu detekce kybernetických útoků. Z tohoto důvodu byl zvolen následující postup. V závislosti na stochastické povaze využívaných algoritmů byla každá z vybraných variant kombinace hyperparametrů otestována. Toto testování se v každém případě opakovalo stokrát, podobně jako v kapitole „Porovnání algoritmu strojového učení při základním nastavení“. Výsledky jsou prezentovány ve dvou typech grafů. V první je využit sloupcový graf vyjadřující Friedmanův test včetně Nemenyiho kritické vzdálenosti. Tento typ grafu zahrnuje všechny využité metriky pro hodnocení klasifikačních modelů. Tato prezentace výsledků vyjadřuje celkový pohled na algoritmus prostřednictvím více metrik. V rámci této prezentace výsledků však zaniknou významnosti jednotlivých metrik. Proto výsledky jsou prezentovány i pomocí krabicových grafů. V těchto grafech jsou zobraceny výsledky nejdůležitější metriky M_{FPR} pro systémy ICS.

Pro jednotlivé datasety byly postupně využity jednotlivé algoritmy strojového učení při základním nastavení, popřípadě při nastavení pomocí optimalizačních algoritmů. U každé kombinace je vypočtena testovací statistika a p-hodnota. Následuje porovnání jednotlivých zástupců pomocí Friedmanova testu včetně Nemenyiho kritické vzdálenosti. Z důvodu velkého počtu dat byla finální statistika a podrobné výsledky (Friedmanova testu včetně Nemenyiho kritické vzdálenosti) umístěny do **Přílohy E**. V každém grafu je diskutováno 12 hodnot, odpovídajících jednotlivým kombinacím od data1 do data12. Kde data1, data2 a data3 představují základní nastavení algoritmů v pořadí neuronová síť, LSTM, IF. Data4 až data6 reprezentují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí evolučního algoritmu. Data7 až data9 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí RS. Data10 až data12 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí TPE.

Diskuse dílčích výsledků

Obsah probírané kapitoly byl zaměřen na prezentaci výsledků jednotlivých algoritmů strojového učení, jejichž hyperparametry byly vybrány podle optimalizačních algoritmů. Lze konstatovat, že ve většině případů byl zjevný rozdíl mezi jednotlivými algoritmy strojového učení při detekci kybernetických útoků. Pro analýzu výsledků byl zvolen následující postup. Každý detekční algoritmus byl v rámci každého kybernetického útoku označen podle jeho pořadí

prostřednictvím Friedmanova testu (podle pořadí). Výsledná suma vyjadřuje konečné skóre popisující výsledná pořadí algoritmů. Tudíž algoritmus s nejnižším skóre byl v tomto případě ta nejlepší varianta. Souhrnné výsledky jsou zobrazeny v Tab. 29. Zde jsou zohledněny všechny výsledky, které přináleží třem datasetům. Kde data1, data2 a data3 představují základní nastavení algoritmů v pořadí neuronová síť, LSTM, IF. Data4 až data6 reprezentují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí evolučního algoritmu. Data7 až data9 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí RS. Data10 až data12 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí TPE.

Nejlepšími zástupci z výsledné tabulky jsou následující algoritmy: neuronová síť nastavená podle RS a neuronová síť nastavená podle evolučního algoritmu. Tyto výsledky se však nedají považovat za jednoznačné, a to především proto, že v rámci prezentovaných grafů jsou zobrazené algoritmy porovnávány podle pěti metrik, přičemž v tomto případě jsou všechny metriky stejně významné. Tedy žádná metrika nevybočuje svojí významností nad ostatní. Takto specifikovaná analýza nám dává ucelený obrázek o detekčních schopnostech algoritmů strojového učení podle všech ukazatelů (metrik). Tento pohled na výsledky však nemůže být jediný, jelikož by v něm zanikl význam významnějších metrik (jako je M_{FPR}).

Tab. 29 – Komparace jednotlivých algoritmů podle pořadí v rámci Friedmanova testu. [vlastní zdroj]

Zastoupené algoritmy/ datasety	**data1**	**data2**	**data3**	**data4**	**data5**	**data6**	**data7**	**data8**	**data9**	**data10**	**data11**	**data12**
Dataset 1	4	2	10	3	5	11	1	6	7	1	8	9
	5	2	10	4	9	7	3	7	6	1	11	8
	6	5	11	2	4	7	3	7	10	1	9	8
	3	1	9	5	6	12	4	8	11	2	7	10
Dataset 2	2	3	3	8	7	2	1	6	5	4	9	2
	1	7	10	8	5	4	3	11	9	6	2	12
	3	2	4	6	6	9	9	1	5	10	7	8
	1	4	8	4	6	7	2	11	9	10	3	5
	10	4	2	8	9	5	6	12	1	7	11	3
	1	5	9	4	6	6	3	7	10	9	2	8
Dataset 3	5	10	2	1	8	6	3	9	11	4	1	7
	9	10	2	5	11	4	7	3	12	1	8	6
	6	4	1	7	4	4	9	6	2	8	3	5
	6	9	10	1	7	7	5	3	11	2	4	8
	1	6	8	3	7	11	2	9	10	5	4	12
	10	7	3	1	6	8	4	11	9	2	5	7
Suma	73	81	102	70	106	110	65	117	128	73	94	118

Analýza metriky M_{FPR} u dílčích variant algoritmů

V této sekci jsou analyzovány výsledky jednotlivých algoritmů strojového učení na základě velmi významné metriky M_{FPR}, která je speciálně důležitá pro systémy ICS. Výsledky byly generovány na základě výsledků (kombinace hyperparametrů) vzniklých prostřednictvím uskutečněných experimentů v metacentru, kde byly využity optimalizační algoritmy. Výsledné kombinace hyperparametrů byly využity pro vygenerovaní výsledků, kde bylo vygenerováno sto průběhů v rámci každé kombinace pro všechny datasety. Z těchto výsledků jsou extrahována data týkající se parametru M_{FPR}. Proto je využito krabicových grafů pro zobrazení získaných výsledků.

Krabicové grafy představují názorný statistický popis a vizualizaci numerických dat. Tzv. „krabice“ (většinou obdélníkový tvar) v rámci grafu představuje rozložení dat. Horní ohraničení představuje třetí kvartil a spodní ohraničení představuje první kvartil. Dohromady hodnoty spadající do „krabice“ odpovídají 50 % celkových zobrazených hodnot. Vodorovná čára v rámci tohoto grafu představuje medián a značka x představuje aritmetický průměr. Tzv. „vousy“ zobrazují vzdálenější hodnoty od mediánu, které přesahují hodnoty spadající do třetího a prvního kvartilu. Tečky v rámci grafu zobrazují odlehlé hodnoty, které by zkreslovaly statistiku charakteristik, kdyby byly zahrnuty. Podrobné výsledky ve formě krabicových grafů jsou umístěny v **Příloze F**.

Diskuse dílčích výsledků

V rámci této sekce byly představeny výsledky metriky M_{FPR} v podobě krabicových grafů pro jednotlivé algoritmy a datasety. Tato metrika by se dala klasifikovat jako stěžejní pro systémy ICS. Jak již bylo diskutováno v kapitole 1.3, tak právě falešné poplachy skýtají hrozbu pro kontinuální provoz ICS systémů. Právě nepřerušený provoz těchto systémů je nutný a zásadní pro naplnění kritických potřeb společnosti a KI. Z tohoto důvodu byly v této sekci diskutovány výsledky M_{FPR} pro dílčí kybernetické útoky.

Samotné výsledky jsou rozčleněny do tří oblastí. Každá z těchto oblastí koresponduje jednomu ze tří datasetů. U prvního datasetu lze identifikovat zástupce algoritmu strojového učení, který má nejlepší hodnotu metriky M_{FPR} ze všech kybernetických útoků. V tomto případě se ukázalo, že využití optimalizačního algoritmu TPE pro nastavení hyperparametrů neuronové sítě vykazuje nejlepší výsledky. Nelze však opomenout neuronovou síť nastavenou pomocí výsledků získaných prostřednictvím evolučního algoritmu, neuronovou síť nastavenou pomocí optimalizačního algoritmu RS a standardně nastavený algoritmus LSTM. Tito zástupci algoritmů strojového učení dosahují poměrně dobrých výsledků i v porovnání s nejlepším zástupcem algoritmu strojového učení. Z těchto výsledků vyplývá poměrně malá závislost na vybraném optimalizačním algoritmu, jelikož všechny optimalizované neuronové sítě vykazují velmi dobré výsledky v rámci datasetu 1.

V případě druhého datasetu lze registrovat algoritmus IF nastavený podle výsledků získaných pomocí evolučního algoritmu, který má zásadně lepší výsledky oproti všem ostatním algoritmům. Tento zástupce dosahuje pro všechny kybernetické útoky téměř nulovou hodnotu M_{FPR}.

Výsledky související s třetím datasetem naznačují dominanci dvou zástupců algoritmů. Jedná se o algoritmy: IF nastavený pomocí evolučního algoritmu a algoritmu IF, který je nastaven pomocí TPE. U všech kybernetických útoků mají tyto algoritmy téměř totožné výsledky. V případě kybernetického útoku CA3_4 a CA3_5 se vyskytly anomální hodnoty. Tyto výsledky jsou zapříčiněny nulovou hodnotou pozitivní třídy (True positive – nebyl identifikován žádný datový záznam o kybernetickém útoku). Tato hodnota představuje však problém z pohledu kalkulace dílčích metrik jako je M_{FPR}, jelikož se vyskytuje ve výpočtu těchto metrik. Proto výsledky ve zmíněných případech nabývají poměrně velkých hodnot, přitom jejich reálná hodnota M_{FPR} je nulová. V ostatních případech (kybernetické útoky) byla hodnota metriky M_{FPR} téměř nulová.

Z obecného pohledu na tuto sekci lze konstatovat následující závěry. V rámci všech datasetů nelze identifikovat jedno řešení, které by bylo vhodné pro všechny z prezentovaných případů. Ukazují se rozdíly mezi datasety, a tedy i to, že žádné z řešení není nejlepší volbou pro všechny datasety. Popisovaný rozdíl je zvláště patrný mezi prvním datasetem a druhým, třetím datasetem. Tento rozdíl může být zapříčiněn odlišným původem těchto datasetů. Dataset 1 je pořízen pomocí simulací, zatímco dataset 2 a dataset 3 představuje záznam projevů fyzicky vytvořených systémů ICS.

6.2.3 Ověření systému detekce anomálií

Tato sekce je zaměřena na ověření vytvořeného systému detekce. Pro tento účel bylo vybráno šest kybernetických útoků. Tři z datasetu 2 (CA2_7, CA2_8, CA2_9) a tři z datasetu 3 (CA3_7, CA3_8, CA3_9). Tyto kybernetické útoky byly využity primárně pro experimenty vztahující se k ověření vytvořeného systému. Z tohoto důvodu nebyly využity v předchozích sekcích disertační práce. Tato separace zajišťuje nezávislost výsledků experimentů této sekce, jelikož neexistuje spojení mezi vytvořeným systémem detekce anomálií a popsanými kybernetickým útoky.

Z důvodu stochastické povahy využívaných algoritmů strojového učení byl zvolen následující postup experimentů. Pro každou variantu kombinace algoritmů strojového učení a optimalizačních algoritmů (9 variant) bylo provedeno sto opakování tvorby prediktivního modelu včetně jejich ohodnocení. Varianty algoritmů strojového učení při standardním nastavení hyperparametrů nebyly v této sekci využity z důvodu jejich poměrně horších výsledků z předešlé kapitoly. Výsledky jsou prezentovány v rámci dvou typů grafů. Jako první je využit sloupcový graf vyjadřující Friedmanův test včetně Nemenyiho kritické

vzdálenosti. Tento typ grafu zahrnuje všechny využité metriky pro hodnocení klasifikačních modelů. Tato prezentace výsledků vyjadřuje celkový pohled na algoritmus prostřednictvím více metrik. V rámci této prezentace výsledků však zaniknou významnosti jednotlivých metrik, proto výsledky jsou prezentovány i pomocí krabicových grafů. Prostřednictvím těchto grafů jsou zobraceny výsledky nejdůležitější metriky M_{FPR} pro systémy ICS.

Byly postupně využity jednotlivé algoritmy strojového učení při nastavení pomocí optimalizačních algoritmů pro jednotlivé datasety (dataset 2 a dataset 3). U každé kombinace je vypočtena testovací statistika a p-hodnota. Následuje porovnání jednotlivých zástupců pomocí Friedmanova testu včetně Nemenyiho kritické vzdálenosti. Z důvodu velkého počtu dat byla finální statistika a podrobné výsledky (Friedmanova testu včetně Nemenyiho kritické vzdálenosti) umístěny do **Přílohy G**. V každém grafu je posuzováno 9 hodnot odpovídajících jednotlivým kombinacím od data1 až do data9. Kde data1, data2 a data3 reprezentují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí evolučního algoritmu. Data4 až data6 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí RS. Data7 až data9 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí TPE.

Diskuse dílčích výsledků

Tato kapitola byla zaměřena na ověření výsledků a detekčních schopností vytvořeného systému pro detekci anomálií. Kompletní výsledky této kapitoly jsou součástí **Přílohy G**. Ve třech případech byl nalezen zjevný rozdíl mezi jednotlivými algoritmy strojového učení při detekci kybernetických útoků.

Pro analýzu výsledků byl zvolen následující postup. Každý detekční algoritmus byl v rámci každého kybernetického útoku označen podle jeho pořadí prostřednictvím Friedmanova testu (podle pořadí). Výsledná suma vyjadřuje konečné skóre popisující výsledné pořadí algoritmu. Tudíž algoritmus s nejnižším skóre byl v tomto případě ta nejlepší varianta. Souhrnné výsledky jsou zobrazeny v Tab. 30. Zde jsou zohledněny všechny výsledky, které přináleží dvěma datasetům. Hodnoty v tabulce jsou prezentovány v pořadí pro dataset 2 (CA2_7, CA2_8, CA2_9) a pro dataset 3 (CA3_7, CA3_8, CA3_9). Kde data1, data2 a data3 představuje nastavení pomocí evolučního algoritmu v pořadí neuronová síť, LSTM, IF. Data4 až data6 reprezentují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí optimalizačního algoritmu RS. Data7 až data9 představují algoritmy neuronová síť, LSTM, IF při nastavení hyperparametrů pomocí TPE.

Mezi nejlepší zástupce z výsledné tabulky lze zařadit následující algoritmy: algoritmus IF nastavená prostřednictvím evolučního algoritmu (data3) – skóre 10 a algoritmus LSTM nastaven podle evolučního algoritmu (data2) – skóre 19. Tyto

výsledky jsou poměrně zajímavé, zvláště pak při srovnání s výsledky v Tab. 30. Algoritmus IF, který má vybrané hyperparametry podle evolučního algoritmu zde ukazuje své dominantní umístění v porovnání se zbylými variantami algoritmů strojového učení.

Tyto výsledky se však nedají považovat za jednoznačné, a to především proto, že v rámci prezentovaných grafů jsou zobrazené algoritmy porovnávány podle pěti metrik, přičemž v tomto případě jsou všechny metriky stejně významné. Tedy žádná metrika nevybočuje svojí významností nad ostatní. Takto specifikovaná analýza nám dává ucelený obrázek o detekčních schopnostech algoritmů strojového učení podle všech ukazatelů (metrik). Tento pohled na výsledky však nemůže být jediný, jelikož by v něm zanikl význam významnějších metrik (jako je M_{FPR}).

Tab. 30 – Komparace jednotlivých algoritmů podle pořadí v rámci Friedmanova testu (ověření výsledků). [vlastní zdroj]

Zastoupené algoritmy/ datasety	**data1**	**data2**	**data3**	**data4**	**data5**	**data6**	**data7**	**data8**	**data9**
Dataset 2	1	3	2	4	7	5	6	2	5
	3	4	1	2	5	6	7	2	6
	7	2	1	3	1	6	4	5	8
Dataset 3	1	5	3	2	6	7	4	8	5
	8	2	2	6	5	4	7	3	1
	4	3	1	6	9	8	7	5	2
Suma	24	19	10	23	33	36	35	25	27

Porovnání metriky M_{FPR} pro jednotlivá řešení vzhledem k ověření systému detekce anomálií

V této sekci jsou ověřena nastavení hyperparametrů pro jednotlivé algoritmy strojového učení na šesti kybernetických útocích. Ověření je provedeno na základě velmi významné metriky M_{FPR}, která je velmi důležitá pro systémy ICS. Výsledky byly generovány na základě výsledků (kombinace hyperparametrů) vzniklých prostřednictvím uskutečněných experimentů v metacentru, kde byly využity optimalizační algoritmy. Výsledné kombinace hyperparametrů byly využity pro vygenerovaní výsledků, kde bylo vygenerováno sto průběhů v rámci každé kombinace pro dva datasety (dataset 2 a dataset 3). Bylo vybráno šest kybernetických útoků (CA2_7, CA2_8, CA2_9, CA3_7, CA3_8, CA3_9) spadajících do těchto datasetů. V rámci vzniklých výsledků byla extrahována

data týkající se parametru M_{FPR}. Z tohoto důvodu je využito krabicových grafů pro zobrazení získaných výsledků.

Krabicové grafy představují názorný statistický popis a vizualizaci numerických dat. Tzv. „krabice“ (většinou obdélníkový tvar) v rámci grafu představuje rozložení dat. Horní ohraničení představuje třetí kvartil a spodní ohraničení představuje první kvartil. Hodnoty spadající do „krabice“ odpovídají 50 % celkových zobrazených hodnot. Vodorovná čára v grafu představuje medián a značka x představuje aritmetický průměr. Tzv. „vousy“ zobrazují vzdálenější hodnoty od mediánu, které přesahují hodnoty spadající do třetího a prvního kvartilu. Tečky v rámci grafu zobrazují odlehlé hodnoty, které by zkreslovaly statistiku charakteristik, pokud by byly zahrnuty. Podrobné výsledky ve formě krabicových grafů jsou umístěny v **Příloze H**.

Diskuse dílčích výsledků

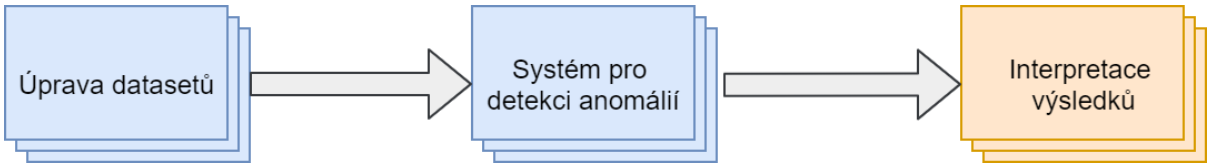
Ověření výsledků daného systému detekce anomálií z pohledu metriky M_{FPR} byla hlavní náplní této sekce. Všechny výsledky v podobě krabicových grafů jsou obsaženy v **Příloze H**. Metrika M_{FPR} by se dala klasifikovat jako stěžejní pro systémy ICS. Jak již bylo diskutováno v kapitole 1.3, tak právě falešné poplachy skýtají hrozbu pro kontinuální provoz ICS systémů. Kontinuální provoz těchto systémů je kritický a zásadní pro naplnění kritických potřeb společnosti a KI. Z tohoto důvodu byly v této sekci diskutovány výsledky M_{FPR} pro dílčí kybernetické útoky.

Samotné výsledky jsou rozčleněny do dvou oblastí. Každá z těchto oblastí koresponduje jednomu ze dvou datasetů (dataset 2 a dataset 3). Výsledky v podstatě kopírují trend, kdy byly využity kybernetické útoky pro tvorbu systému detekce anomálií (např. CA2_1, CA2_2). V případě druhého datasetu byl opět dominantní algoritmus IF, který byl nastaven pomocí evolučního algoritmu ve všech případech (CA2_7, CA2_8, CA2_9). U třetího datasetu byly identifikování dva nejlepší zástupci, kteří mají obdobné výsledky. Jedná se o algoritmus strojového učení IF, který je v prvním případě nastaven pomocí evolučního algoritmu a ve druhé případě pomocí optimalizačního algoritmu TPE.

Z obecného pohledu na tuto sekci lze konstatovat následující závěry. Souhrnně lze identifikovat nejlepšího zástupce pro dva vybrané datasety (dataset 2 a dataset 3). V tomto případě se jedná o algoritmus strojového učení IF, jehož hyperparametry byly nastaveny podle výsledků vzešlých z evolučního algoritmu. Ve všech případech tento zástupce dosahuje výborných výsledků, kde metrika M_{FPR} dosahuje velmi nízkých až nulových hodnot. Takto nastavený systém má poté v reálném provozu velmi nízké procento falešných poplachů.

6.2.4 Interpretace anomálií detekovaných pomocí algoritmů strojového učení

Tato kapitola je zaměřena na zhodnocení možnosti interpretace výsledků. Výsledky kapitoly fundamentálně vycházejí z předešlých kapitol disertační práce, a tedy na ně i navazují. V popisované kapitole jsou analyzovány možnosti interpretace představeného detekčního systému. Z ohledu aplikace každého detekčního systému je vhodné definovat i možnosti automatické interpretace. A to především z pohledu rychlé reakce na vzniklou mimořádnou situaci, kde hledisko času hraje velmi důležitou roli.



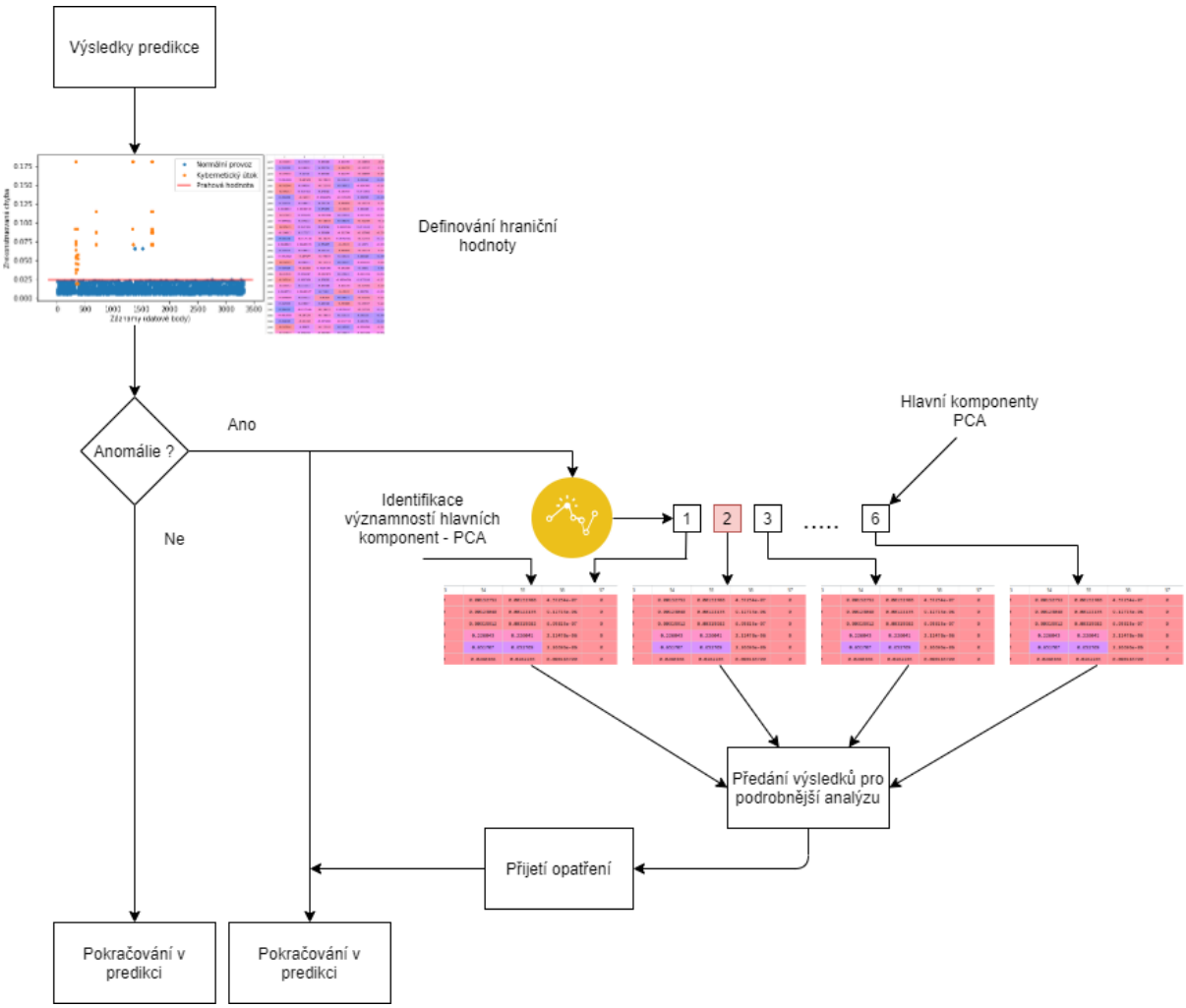
Obr. 44: Diagram procesů pro tvorbu algoritmu – interpretace výsledků. [vlastní zdroj]

V předešlých experimentech byli identifikováni dva nejlepší zástupci (algoritmy strojového učení včetně nastavení hyperparametrů) pro detekci kybernetických útoků. Jedná se o neuronovou síť, jejichž hyperparametry jsou nastaveny na základě optimalizačního algoritmu TPE a algoritmus strojového učení IF, jenž má nastavené hyperparametry podle výsledků získaných pomocí genetického algoritmu. Tyto dvě zvolené varianty se však od sebe mírně odlišují z pohledu interpretace výsledků. Oba algoritmy využívají podobnou strukturu dat, i když využívají rozdílné transformace ve smyslu náhrady prázdných hodnot a změny měřítka. Řešení kategorických dat a s tím spojené postupy pro redukci dimensionality jsou u obou algoritmů totožné. Z tohoto pohledu je postup interpretace výsledků u obou algoritmů téměř stejný.

Prvním krokem k interpretaci výsledků je shromáždění informací, týkajících se predikcí jednotlivých datových bodů. Tento krok byl již proveden v případě neuronové sítě, popřípadě LSTM algoritmu. Kde je již v rámci predikce vypočtena hodnota abnormality pro každý datový bod v rámci každého atributu (rozdíl reálných a predikovaných dat). Tento výpočet je však v případě IF (podle získaných znalostí) nemožný. Zejména z důvodu náhodného procesu pro rozdělování na každém z uzlů. Z tohoto důvodu je nutné zvolit mírně odlišný postup. Musí být vytvořen druhý model algoritmu strojového Random forest (RF) učení k modelu IF. RF poté dostává výstup z IF, pokud byla identifikována anomálie. RF poté může identifikovat jednotlivé významnosti datových bodů.

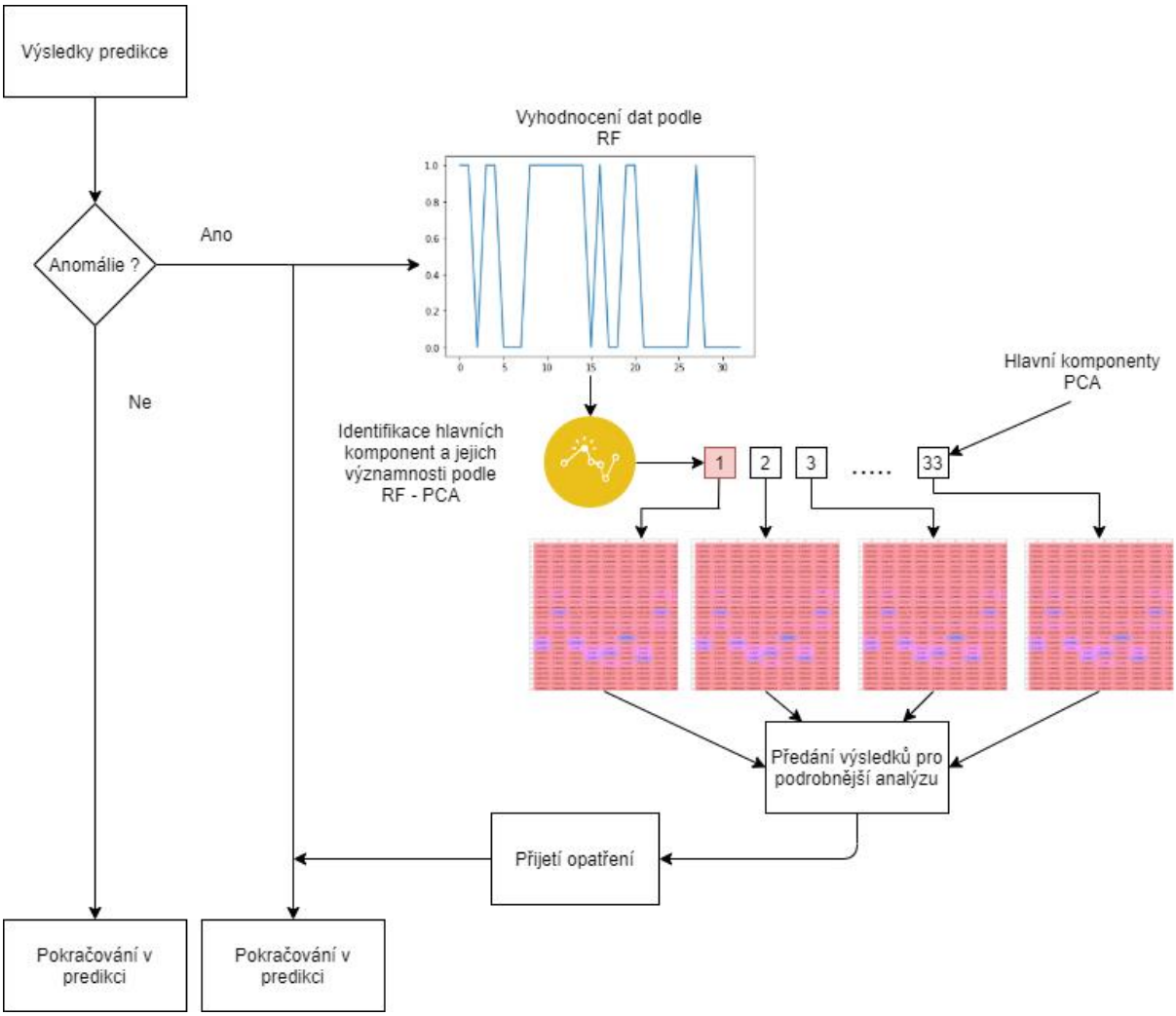
V následujících odstavcích jsou popsány využité metody pro interpretaci výsledků. V Obr. 45 je zobrazen diagram postupu pro interpretaci výsledků pro neuronovou síť. Celý proces začíná získáním predikce výsledků pomocí modelu neuronové sítě. Následuje proces výpočtu hraniční hodnoty pro rozdělení

anomálního chování od normálního provozu systému. Celý proces určení a výpočtu hraniční hodnoty byl již popsán v kapitole 4.4.2. V závislosti na hraniční hodnotě je možné identifikovat anomální hodnoty. Jestli není nalezena anomálie poté provoz detekčního systému může pokračovat stejně i nadále. Pokud je však anomálie nalezena, pak je nutné ji identifikovat a interpretovat. Jak je v diagramu názorně ukázáno definice hraničních hodnot se děje hned po predikci. Tento proces je vlastně součástí detekce anomálií. Lze také konstatovat, že v této fázi interpretace jsou již známé jednotlivé hodnoty pro dílčí datové body. Jedná se o rozdíl reálné hodnoty a predikované hodnoty pro každý bod. Při identifikaci anomálie následuje poté poměrně jednoduchý postup. Tedy identifikace anomálního datového bodu hlavní komponenty, která k němu přináleží. V prezentovaném případě se jedná o hlavní komponentu 2 (viz diagram). Poté reverzní metodou lze docílit výčtu atributů v rámci této hlavní komponenty včetně jejich významnosti. Tedy jak významný je každý z dílčích atributu pro sestavení hlavní komponenty (PCA – kapitola 4.2.1). Prostřednictvím tohoto postupu lze zprostředkovaně vypočítat významnost jednotlivých atributů.



Obr. 45: Interpretace výsledků neuronové sítě – dataset 2. [vlastní zdroj]

Výsledky takto získané mohou být podrobeny detailnější analýze, z které by následně měla být vytvořena nezbytná opatření pro úpravu chodu sledovaného systému, popřípadě přijetí úprav detekčního systému v případě falešné identifikace. V případě pozitivní identifikace kybernetického útoku by měla být přijata taková opatření, která zabraňují v pokračování útoku nebo zmírňují jeho dopady. Tato disertační práce nebyla svým pojetím koncipována, tak aby tyto body vyřešila. V řadě případů je nutná hluboká znalost chráněného systému a jeho procesů pro efektivní analýzu a také přijetí opatření z toho vycházející. V těchto případech je poté nutná úzká koordinace s technickými pracovníky chráněného systému pro vyřešení této problematiky.



Obr. 46: Interpretace výsledků algoritmu IF – dataset 1. [vlastní zdroj]

Ve druhém případě (Obr. 46) je k identifikaci anomálií využit algoritmus strojového učení IF. Jak již bylo diskutováno v předešlých odstavcích, tak pro interpretaci výsledků v tomto případě bylo zapotřebí využít algoritmus RF. V tomto případě identifikaci anomálií provádí algoritmus IF, poté co je rozhodnuto, jestli se jedná o anomálii nebo ne. Nejedná-li se o anomálii, tak se pokračuje v predikci beze změny. Pokud se jedná o anomálii, tak je provedena

interpretace nalezené anomálie. Jak již bylo diskutováno, tak k tomu je zapotřebí využít algoritmu RF. Jako vstupní data jsou tomuto algoritmu poskytnuty výstupy z předešlého algoritmu (IF). Tedy dataset, kde jsou data již klasifikována podle algoritmu IF. Zde je ke každému záznamu přiřazeno označení, jestli patří mezi anomálie nebo do oblasti normálního provozu. Algoritmus RF poté vypočítá každému datovému bodu (každé hlavní komponentě) výsledný vliv. Tato data jsou následně využita pro selekci významných datových bodů od bezvýznamných na základě aritmetického průměru po významné. Z grafu na Obr. 46 je patrné binární rozdělení datových bodů na významné a nevýznamné. Další postup je poté obdobný jako v předešlém případě. Jedinými rozdíly jsou: rozdílná dimense výsledných atributů (rozdílný dataset) a významná hlavní komponenta (v tomto případě se jedná o první hlavní komponentu).

Diskuse dílčích výsledků

Obsahem této kapitoly bylo představení možností interpretace výsledků systému detekce anomálií. Proces interpretace byl představen na dvou případech. V prvním případě byla interpretace založena na neuronové síti využívající již představený postup pro detekci anomálií (viz. kapitola 5.4.2). Ve druhém případě byl využit algoritmus strojového učení RF. Zde muselo být využito paralelního nasazení algoritmu IF pro účel interpretace anomálií. Na obou prezentovaných případech byly demonstrovány možnosti interpretace prostřednictvím identifikace významných atributů v rámci každé detekované anomálie. Tyto získané informace poté mohou být využity pro další analýzu typu a původu detekovaného kybernetického útoku. Interpretace výsledků také umožňuje rychlejší identifikaci anomálií způsobenou neúmyslně, jako je technické selhání systému nebo selhání systému kvůli lidskému faktoru. V neposlední řade je proces interpretace využit jako zpětná vazba pro systém detekce anomálií. Kde je pomocí interpretace nalezena příčina falešných poplachů (falešně pozitivní klasifikace), na základě, které je možné upravit klasifikační model pro potřeby detekce anomálií.

7. PŘÍNOS PRO VĚDU A PRAXI

V rámci předložené disertační práce byl proveden výzkum v oblasti detekce anomálií pomocí algoritmů a metod strojového učení v oblasti ICS. Tato kapitola je zaměřena na popsání přínosů pro vědu a praxi, které vycházejí z provedeného výzkumu při řešení disertační práce.

7.1 PŘÍNOS PRO VĚDU

V rámci předložené disertační práce bylo realizováno značné množství experimentů z oblasti strojového učení, které definují a validují navržený postup ve formě vytvořeného systému pro detekci anomálií. Výstupy výzkumu v rámci disertační práce najdou své uplatnění v oblastech strojového učení, detekce anomálií, kybernetické ochrany, optimalizace, metod matematické statistiky a dalších příbuzných vědeckých oblastech.

Hlavním cílem disertační práce bylo vytvoření systému pro detekci anomálií v oblasti ICS. Tento systém byl od počátku koncipován s ohledem na specifika ICS systémů, při zachování možnosti interpretace výsledků. Tyto charakteristiky spolu s určitou adaptabilností navrhovaného řešení lze považovat za jeden z hlavních přínosů pro vědeckou komunitu. K dosažení tohoto cíle bylo zapotřebí multidisciplinárního přístupu, ve kterém se spojují znalosti spadající do oblasti kybernetické bezpečnosti, umělé inteligence, detekce anomálií, dolování dat, úpravy datasetů, optimalizace, multikriteriálního hodnocení, ale i oblasti průmyslových řídicích systémů. V rámci těchto definovaných oblastí lze očekávat přínosy pro vědeckou komunitu.

Jako další přínos pro vědu je potvrzení aplikovatelnosti představeného systému detekce anomálií i pro reálné systémy. Tedy aplikovatelnost tohoto systému v reálném prostředí při využití reálných dat. Využití detekce anomálií na základě algoritmů strojového učení se ukázalo jako velmi slibná oblast, která povede ke zvýšení kybernetické bezpečnosti i velmi komplexních systémů.

Vývoj představeného systému pro detekci anomálií lze rozdělit do pomyslných čtyř sekcí. První z nich je zaměřena na oblast mapování kybernetických hrozeb pro systémy ICS. V rámci této oblasti byla vytvořena metoda pro nalezení nejzávažnějších hrozeb založená na dolování databází a nástroji Shodan. Následující sekce jsou zaměřeny na tvorbu uceleného nástroje pro detekci anomálií, které jsou klasifikovány jako kybernetické útoky. V rámci každé sekce byla uskutečněna řada experimentů pro jednotlivé algoritmy, techniky a vybraná nastavení. Druhá sekce zahrnuje výzkum spojený s datovou úpravou vstupních dat, které jsou využity pro detekci anomálií v systémech ICS. Provedené experimenty definují využité techniky a jejich způsobilost pro nasazení v oblasti detekce anomálií pro kybernetickou bezpečnost ICS lze považovat za další přínos pro vědeckou komunitu.

Využití algoritmů strojového učení ve spojení s optimalizačními algoritmy představuje hlavní náplň třetí sekce. Každé ze zvolených řešení je optimalizováno s přihlédnutím ke specifikám ICS systémů. Toho bylo dosaženo pomocí aplikace multikriteriálního hodnocení pro řešení vícekriteriální optimalizace. Tento nový postup v oblasti detekce anomálií je dalším přínosem pro vědu.

Výsledky disertační práce naznačují vhodné kombinace optimalizačních algoritmů, algoritmů strojového učení a technik pro úpravu dat v oblasti detekce kybernetických útoků. Z výsledků lze také vyvodit nevhodnost využití některých algoritmů strojového učení pro řešení zadané problematiky. Jedná se zejména o algoritmus OCSVM, který je často využíván odbornou komunitou pro detekci kybernetických útoků v rámci ICS systémů.

V rámci výsledného řešení bylo využito multikriteriálního hodnocení ve spojení s optimalizačními algoritmy pro nalezení nejlepšího nastavení hyperparametrů algoritmů strojového učení. Výstupy naznačují velmi dobré výsledky, zvláště pak v oblasti minimalizace metriky M_{FPR}. Tyto postupy mohou být využity odbornou komunitou pro zvýšení efektivnosti v oblasti detekování kybernetických útoků a detekování anomálií.

Poslední třetí sekce přínosu pro vědeckou komunitu je zaměřena na interpretaci výsledků systému pro detekci anomálií. Při postupu interpretace významnosti jednotlivých atributů je využito reverzních postupů, které se z části odvíjejí od technik využitých v první sekci (techniky pro úpravu dat). Navržený postup lze aplikovat na řadu oblastí strojového učení, kde je využito různých vstupních dat o rozdílné dimensionalitě.

7.2 PŘÍNOS PRO PRAXI

Kybernetická bezpečnost kritické infrastruktury se stala jednou z hlavních bezpečnostních otázek současnosti. Navržený systém pro detekci anomálií v oblasti ICS je vytvořen s ohledem na aplikaci v reálném prostředí. Jednotlivé zvolené postupy jsou tudíž zvoleny takovým způsobem, jenž umožnuje nasazení popisovaného řešení v rozličných ICS systémech. Tento postup umožňují optimalizační techniky. Ty jsou využity pro výběr vhodného nastavení (hyperparametrů) algoritmů strojového učení. Toto nastavení i díky využitému multikriteriálnímu hodnocení umožňuje určité adaptace pro každý dílčí systém ICS. Samotný systém pro detekci anomálii je složen ze čtyř částí. První část je zaměřena na identifikaci kritických hrozeb pro oblast ICS ve formě kyberneticích útoků. Takto identifikované útoky by měly být využity ve druhé a třetí části systému. Tedy pro úpravu vstupních dat (část dvě) a pro optimalizaci algoritmu strojového učení (část tři). Poslední částí systému pro detekci anomálií je interpretace výsledků detekce kybernetických útoků (část čtyři).

Většina řešení pro ochranu před kybernetickými útoky je řešena na bázi detekce pomocí signatur nebo metod, které vyžadují přesně definovanou

množinu kybernetických útoků, proti kterým je poté dotyčný systém chráněn. Navrhované řešení popsané v disertační práci, však využívá informace získané v rámci normálního chodu sledovaného systému. Na základě tohoto chování je vytvářen prediktivní model, který je dále využit pro detekci kybernetických útoků. Tento rozdíl oproti klasickému pojetím detekce kybernetických útoků pomocí signatur umožňuje systému detekce anomálií zásadní výhody při detekci kybernetických útoků. Systém detekce anomálií nepotřebuje žádnou databázi (signatur) kybernetických útoků. Takový systém umožňuje minimalizovat náklady spojené s periodickými aktualizacemi databází (signatur) kybernetických útoků. Druhým významným přínosem tohoto představeného řešení je detekce dosud neznámých kybernetických útoků. V případě detekce pomocí signatur tato událost nastává ve dvou případech, při pozdní aktualizaci databáze signatur nebo v případě kdy kybernetický útok vůbec nebyl identifikován odbornou komunitou (případ „Zero Day Attack“). Navržený systém detekce anomálií byl od počátku vytvářen, aby nebyl negativně ovlivněn těmito dvěma případy. Aplikace takto navrženého systému detekce anomálií v praxi nejenom zvýší efektivnost detekce kybernetických útoků, ale také sníží náklady na provoz kybernetické ochrany ICS systému.

Z důvodu ověření a validace systému detekce anomálií byl uskutečněn rozsáhlý výzkum. Byla provedena řada experimentů, která se především zaměřila na optimalizaci algoritmů strojového učení prostřednictvím pěti metrik M_{F1}, M_{MCC}, M_{Prec}, M_{FPR}, Čas. Přičemž právě metrika M_{FPR} (vyjadřuje falešné detekce kybernetických útoků) je nejdůležitější metrikou pro praxi. Výsledná řešení vykazují velmi malé až nulové hodnoty metriky M_{FPR}. Tyto výsledky podporují možnost nasazení systému detekce anomálií v reálném provozu. Právě absence falešných klasifikací nezatěžuje chráněný systém falešnými poplachy, které by mohly ohrozit kontinuitu provozu celého systému ICS. Zbylé metriky M_{F1}, M_{MCC}, M_{Prec}, Čas zajištují pomocí multikriteriálního hodnocení TOPSIS vyváženost výsledného modelu. Při využití pouze metriky M_{FPR} by výsledný systém měl nulový výskyt falešných poplachů, avšak by s největší pravděpodobností nedetekoval žádný kybernetický útok. Vyvážená kombinace těchto metrik byla nutná pro optimální nastavení systému detekce anomálií pro potřeby praxe.

Závěrečná část výzkumu v oblasti systému detekce anomálií byla zaměřena na interpretaci detekovaných kybernetických útoků. Výstupem tohoto cíle disertační práce byly dva navržené systémy pro dva typové příklady. V rámci zvoleného řešení byl každý datový bod (záznam v datasetu) interpretován pro všechny atributy. Ke každé detekované anomálii byly v rámci popisovaného systému identifikovány nejdůležitější atributy, které přispěly k detekci kybernetického útoku. Tyto vybrané atributy jsou ukazateli, příznaky anomálie, a tudíž jejich identifikací lze získat informace týkající se původu a charakteru kybernetického útoku. V rámci praxe mohou organizace využívat tuto metodu pro snížení falešných poplachů, které jsou velmi kritické pro ICS systémy.

8. ZÁVĚR

Disertační práce byla zaměřena na ochranu systémů ICS, které se díky digitalizaci a Průmyslu 4.0 stávají nezbytnou součástí moderní společnosti. Tento trend závislosti na systémech ICS bude v budoucnu posilovat. Lze očekávat implementace ICS v řadě sektorů kritické infrastruktury státu. Z dosavadního vývoje v oblasti kybernetické bezpečnosti vyplývá narůstající zájem jak státních, tak privátních aktérů o systémy ICS. Tento trend je zapříčiněn významností daných systémů pro moderní společnost. Z tohoto důvodu je kybernetická bezpečnost ICS systémů nezbytná pro zachování dostupnosti kritických služeb pro obyvatelstvo.

V rámci disertační práce byl představen ucelený systém pro detekci anomálií založený na strojovém učení. Tento systém byl od počátku navrhován tak aby splňoval následující požadavky:

1. Detekování neznámých kybernetických útoků.
2. Škálovatelnost systému.
3. Možnost reálného využití v systémech ICS.
4. Možnost interpretace výsledků.

Tyto čtyři požadavky byly zásadní pro tvorbu navrhovaného systému. Detekce dosud neznámých kybernetických útoků byla prvním požadavkem a předpokladem pro tvorbu robustního systému detekce anomálií. Tento bod byl splněn výběrem vhodného typu algoritmů strojového učení. Využití kombinace učení s učitelem a učení bez učitele zajišťuje tvorbu klasifikačních modelů výhradně prostřednictvím dat normálního provozu sledovaného systému. Každá odchylka od tohoto modelu je poté detekována jako anomálie. Tento postup zajišťuje flexibilitu pro detekci anomálií. Oproti tomu modely založené na učení s učitelem jsou určeny pro specifické, předem determinované kybernetické útoky.

Škálovatelnost detekčního systému byla zajištěna pomocí transformace dat pomocí algoritmu PCA. Tento postup zajišťuje zpracování detekci anomálií pro teoreticky velmi rozsáhlé systémy ICS. Redukce dimenze datasetu byla názorně demonstrována. Všechny získané výsledky byly založeny na tomto postupu. Z tohoto důvodu může mít dataset neomezené množství atributů, které jsou poté redukovány na přijatelnou úroveň. Také byl představen postup úpravy datasetu, jak numerických hodnot, tak kategorických hodnot včetně úpravy prázdných hodnot a úpravy měřítka dat. Prezentované postupy předcházejí samotné detekci kybernetických útoků a tvoří nezbytný prvek zajišťující efektivní nasazení systému pro detekce anomálií.

Třetí bod byl z pohledu počtu experimentů a časového hlediska nejobtížnějším bodem. Bylo potřeba provést velmi velký počet experimentů, které svými

výsledky buď vyvrátily nebo potvrdily prezentovanou tezi o aplikovatelnosti řešení pro reálné aplikace (Teze: Prezentované řešení je aplikovatelné pro reálné nasazení v rámci systémů ICS). V tomto směru bylo vytvořeno hodnotící skóre podle Fullerova trojúhelníku a metody TOPSIS, kde bylo využito pěti hodnotících metrik M_{F1}, M_{MCC}, M_{Prec}, M_{FPR}, Čas. Jako nejdůležitější metriku lze identifikovat M_{FPR}, která reflektuje množství falešně pozitivních klasifikací. Tedy jednoduše řečeno množství normálního provozu klasifikovaného jako kybernetický útok. Tyto falešné klasifikace mohou být velmi kritické pro systémy ICS z důvodu ohrožení kontinuity provozu. Ta je pro řídicí systémy nejdůležitějším hlediskem viz. kapitola 1.2. Ostatní metriky (M_{F1}, M_{MCC}, M_{Prec}, Čas) zajišťují, aby algoritmus byl vyvážený. Přesněji řečeno zajišťuje, aby nenastala možnost, kdy metrika M_{FPR} je na nejnižší úrovni, avšak tento systém prakticky neidentifikuje žádný kybernetický útok, popřípadě je dotyčné řešení časově velmi náročné.

Následující fáze byla zaměřena na nalezení nejvhodnějšího zástupce pro detekci anomálií v systémech ICS. K tomu bylo využito čtyř algoritmů strojového učení (neuronová síť, LSTM a Isolation forest, OCSVM) a tří datasetů. Algoritmu OCSVM byl z této množiny vyloučen, a to především z důvodu špatných výsledků v podstatě ve všech experimentech (viz. Kapitola 5.2.2 a především její dílčí závěr). Bylo nadále otázkou, zdali zbylé algoritmy strojového učení (neuronová síť, LSTM a Isolation forest) nabývají nejlepších možných vlastností a nelze je dále vylepšit. Z tohoto důvodu byly provedeny experimenty s těmito algoritmy ve snaze je optimalizovat prostřednictvím nastavení jejich hyperparametrů. Využito bylo tří optimalizačních algoritmů (genetický algoritmus, RS, TPE), kde jako cílové funkce (OF) optimalizace bylo využito navržené hodnotící skóre. V celkovém počtu bylo provedeno 270 dílčích experimentů, přičemž každý z experimentů probíhal maximálně 300 hodin. Celkově se tedy jednalo o 81 000 hodin strojového času, spotřebovaného k realizaci experimentů.

Výsledky těchto testů jsou prezentovány pomocí Friedmanova testu včetně Nemenyiho kritické vzdálenosti, kde pomocí součtu pozicí v rámci prezentovaného testu je dosaženo celkové pozice jednotlivých algoritmů. Mezi nejlepší zástupce z výsledné tabulky lze zařadit následující algoritmy: neuronová síť nastavená podle RS a neuronová síť nastavená podle evolučního algoritmu. Je nutné mít na paměti, že toto porovnání je provedeno na základě všech pěti hodnotících metrik, proto je jeho vypovídající hodnota omezena. Z tohoto důvodu bylo provedeno druhé porovnání zaměřující se na metriku M_{FPR}. V rámci tohoto porovnání bylo využito krabicových grafů. Výsledky poukazují na rozdílné výsledky pro různé datasety. Respektive rozdílné výsledky pro dataset 1, kde je nejlepší možností varianta neuronové sítě, která má nastavené hyperparametry podle výsledků získaných prostřednictvím optimalizačního algoritmu TPE a výsledky pro dataset 2, 3, kde nejlepších výsledků dosahuje

algoritmus strojového učení IF při nastavení hyperparametrů podle evolučního algoritmu.

Z těchto výsledků lze dedukovat následující závěr: nebyla nalezena jedna varianta algoritmu, která by bylo vhodná pro všechny datasety. Avšak existuje možnost, že tento rozdíl ve výsledcích je zapříčiněn rozdílnou strukturou datasetů mezi datasety 2, 3 a datasetem 1. Datasety 2 a 3 jsou vytvořeny na základě záznamů z fyzicky vytvořených systémů. Zatímco dataset 1 je vytvořen na základě simulací. Tyto závěry jsou s velkou pravděpodobností pravdivé i pro jiné datasety. Ze získaných výsledků byly identifikovány dvě výsledné kombinace, které mají značný potenciál. Zvláště pak řešení pomocí algoritmu IF vykazuje vhodné parametry. Prakticky nulovou metriku M_{FPR} téměř ve všech případech kybernetických útoků v datasetu 2 a 3. Pro nalezení finálního datasetu se tedy doporučují testy chráněného systému před nasazením jednoho z výsledných řešení.

Z pohledu obou výsledných srovnání, tedy podle všech pěti metrik a podle metriky M_{FPR}, vykazuje neuronová síť (TPE) poměrně velmi dobré výsledky v obou případech. Oproti tomu algoritmus IF (evoluční algoritmus), vykazuje nejlepší možné výsledky v případě komparace metriky M_{FPR}. V rámci dvou datasetů, avšak nevykazuje příliš dobré výsledky v rámci srovnání všech pěti metrik. Z toho vyplývají následující závěry pro algoritmus IF. Při prakticky nulové hodnotě metriky M_{FPR} algoritmus IF identifikuje jen velmi omezené množství anomálních bodů kybernetického útoku. Avšak i toto omezené množství stačí k pozitivnímu zachycení dotyčného kybernetického útoku. Dá se tedy konstatovat, že nejdůležitější parametry detekčního systému jsou zachovány, tedy identifikace kybernetického útoku a prakticky nulové množství falešné identifikace. Při budoucí aplikaci tohoto řešení by byla nejspíše vhodná paralelní aplikace systému detekce anomálií s jiným detekčním řešením, které by zvýšilo množství zachycených anomálních záznamů, a to především z důvodu další identifikace a interpretaci zachyceného kybernetického útoku.

Systém detekce anomálií byl ověřen prostřednictvím šesti kybernetických útoků. Tři pro dataset 2 a tři pro dataset 3. Tyto kybernetické útoky nebyly využity pro tvorbu systému detekce anomálií. Tato separace byla učiněna z důvodu nezávislého ověření představeného systému. Byly provedeny dva experimenty, tedy první prostřednictvím porovnání algoritmů pomocí pěti metrik. Druhý experiment byl zaměřen na metriku M_{FPR}. Výsledky obou experimentů ukazují dominantní pozici algoritmu strojového učení IF, který byl nastaven pomocí evolučního algoritmu. Tato varianta vykazovala nejlepší výsledky nejenom v rámci druhého experimentu (porovnání podle metriky M_{FPR}), ale také v prvním experimentu. Tudíž má tato varianta nejlepší detekční schopnosti nejenom ve formě identifikace kybernetických útoků, ale také

negeneruje falešné poplachy. Takto navržený systém detekce anomálií je možné aplikovat v reálném prostředí při kybernetické ochraně systému ICS.

Závěrečná kapitola byla zaměřena na interpretaci detekovaných anomálií. Bylo využito reverzního postupu pro získaní nejvýznamnějších atributů pro každý klasifikovaný anomální záznam. Takto klasifikované výsledky jsou poté předány pro detailnější analýzu, která je následně nezbytná pro tvorbu opatření a úpravu chodu sledovaného systému, popřípadě přijetí úprav detekčního systému v případě falešné identifikace. V případě pozitivní identifikace kybernetického útoku by měla být přijata taková opatření, která zabraňují v pokračování útoku, nebo zmírňují jeho dopady. Tato disertační práce si však svým pojetím neklade za cíl finální interpretaci původu a typu kybernetického útoku, popřípadě detekci útočníka. V řadě případů je nutná hluboká znalost chráněného systému a jeho procesů pro efektivní analýzu a také přijetí opatření z toho vycházející. V těchto případech je poté nutná úzká koordinace s technickými pracovníky chráněného systému pro vyřešení této problematiky.

V rámci prezentovaného systému detekce byl řešen i postup identifikace relevantních kybernetických útoků pro systémy ICS. V podsekci 5.1.1 byla řešena identifikace nezranitelnějších míst ICS systému k čemuž byla využita americká databáze zranitelností ICS-CERT. Na tuto kapitolu navazuje kapitola 5.1.2, ve které byl nastíněn proces využití dataze ICS-CERT a nástroje Shodan pro identifikaci relevantních zranitelností v rámci systému ICS. Z provedeného výzkumu jsou patrné zranitelnosti, které jsou zásadními nedostatky systému ICS, a to je především velmi nízká reakční doba na nové hrozby vyplývající z charakterů systémů ICS.

Prezentovaná disertační práce byla vytvořena se záměrem vytvoření automatizovaného detekčního systému pro kybernetické útoky. Svým tématem byla zaměřena na velmi důležitou oblast systémů ICS spadající i do Průmyslu 4.0. K naplnění tohoto cíle bylo využito řady metod, nástrojů a algoritmů. Z prezentovaných výsledků vyplývají následující poznatky:

1. Představený systém umožňuje detekci neznámých kybernetických útoků.
2. Prezentovaný systém je plně škálovatelný, a tudíž je jej možné využít v rozdílných systémech ICS.
3. Ze souboru výsledků vyplývají dvě potencionální řešení pro ochranu reálných ICS systémů. Tato řešení vykazují velmi nízké hodnoty metriky M_{FPR}.
4. Možnost interpretace výsledků je možný na základě reverzního přístupu k vyhodnoceným výsledkům.

Uskutečněný výzkum byl zaměřen na dynamickou oblast detekce anomálií vztahující se ke kybernetickým útokům. Systém detekce anomálií byl vytvořen a

otestován. Výsledky potvrzují využitelnost systému v reálném prostředí. Hlavně z důvodu velmi nízkého počtu falešných poplachů. Avšak ostatní metriky (M_{F1}, M_{MCC}, M_{Prec}) kromě metriky Čas, nevykazují vysoké hodnoty. Tento dílčí problém je otázkou pro budoucí výzkum. Jedna z možných cest je paralelní využití algoritmů strojového učení pro detekci anomálií, kde výstup by byl založen na formě hlasování využitých algoritmů. Další možnou oblastí budoucího výzkumu je otázka interpretace kybernetických útoků. Tedy vývoj analytického nástroje pro analýzu výsledků systému detekce anomálií. Tento systém by měl být zaměřen na forenzní analýzu detekovaných anomálii. Tyto anomálie by poté (podle analytického systému) byly přesněji a detailněji identifikovány a klasifikovány do jedné ze skupin kybernetických útoků.

9. SEZNAM POUŽITÉ LITERATURY

[1] FRANK, Alejandro Germán; DALENOGARE, Lucas Santos; AYALA, Néstor Fabián. Industry 4.0 technologies: Implementation patterns in manufacturing companies. International Journal of Production Economics, 2019, 210: 15-26.

[2] FALLIERE, Nicolas; MURCHU, Liam O.; CHIEN, Eric. W32. stuxnet dossier. White paper, Symantec Corp., Security Response, 2011, 5.6: 29.

[3] ELLIS, Ryan; MOHAN, Vivek (ed.). Rewired: Cybersecurity Governance. John Wiley & Sons, 2019.

[4] BENSON, Vladlena; MCALANEY, John; FRUMKIN, Lara A. Emerging threats for the human element and countermeasures in current cyber security landscape. In: Cyber Law, Privacy, and Security: Concepts, Methodologies, Tools, and Applications. IGI Global, 2019. p. 1264-1269.

[5] JALALI, Mohammad S., et al. Health care and cybersecurity: bibliometric analysis of the literature. Journal of medical Internet research, 2019, 21.2: e12644.

[6] ALASSAFI, Madini O., et al. Security in organisations: governance, risks and vulnerabilities in moving to the cloud. In: International Workshop on Enterprise Security. Springer, Cham, 2015. p. 241-258.

[7] STOUFFER, Keith; FALCO, Joe; SCARFONE, Karen. Guide to industrial control systems (ICS) security. NIST special publication, 2011, 800.82: 16-16.

[8] MACAULAY, Tyson; SINGER, Bryan L. Cybersecurity for industrial control systems: SCADA, DCS, PLC, HMI, and SIS. CRC Press, 2011.

[9] GINTER, Andrew. SCADA Security-What's broken and how to fix it. Lulu. com, 2019.

[10] ANTON, Simon D. Duque; HAFNER, Alexander; SCHOTTEN, Hans Dieter. Devil in the detail: Attack scenarios in industrial applications. In: 2019 IEEE Security and Privacy Workshops (SPW). IEEE, 2019. p. 169-174.

[11] LI, Pengzhong. Introductory Chapter: New Trends in Industrial Automation. In: New Trends in Industrial Automation. IntechOpen, 2019.

[12] KIM, Dong-Seong; TRAN-DANG, Hoa. An Overview on Industrial Control Networks. In: Industrial Sensors and Controls in Communication Networks. Springer, Cham, 2019. p. 3-16.

[13] KROTOFIL, Marina; KURSAWE, Klaus; GOLLMANN, Dieter. Securing industrial control systems. In: Security and Privacy Trends in the Industrial Internet of Things. Springer, Cham, 2019. p. 3-27..

[14] LUIIJF, Eric; JAN TE PASKE, Bert. Cyber Security of Industrial Control Systems [online]. In: Global Conference on Cyberspace (GCCS), 2015. Dostupné z: http://publications.tno.nl/publication/34616507/KkrxeU/luiijf-2015-cyber.pdf

[15] PATZER, Florian, et al. Towards computer-aided security life cycle management for critical industrial control systems. In: International Conference on Critical Information Infrastructures Security. Springer, Cham, 2018. p. 45-56.

[16] HUMAYED, Abdulmalik, et al. Cyber-physical systems security—A survey. IEEE Internet of Things Journal, 2017, 4.6: 1802-1831.

[17] ANI, Uchenna P. Daniel; HE, Hongmei; TIWARI, Ashutosh. Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective. Journal of Cyber Security Technology, 2017, 1.1: 32-74.

[18] Risidata. Risi [online]. 2015 [cit. 2020-10-21]. Dostupné z: http://www.risidata.com/

[19] KASPERSKY, I. C. S. Threat landscape for industrial automation systems in H1 2018. [online]. 2018 [cit. 2020-10-21]. Dostupné z: https://securelist.com/threat-landscape-for-industrial-automation-systems-in-h1-2018/87913/

[20] TANGE, Koen, et al. Towards a systematic survey of industrial IoT security requirements: research method and quantitative analysis. In: Proceedings of the Workshop on Fog Computing and the IoT. 2019. p. 56-63.

[21] DEWA, Zibusiso; MAGLARAS, Leandros A. Data mining and intrusion detection systems. International Journal of Advanced Computer Science and Applications, 2016, 7.1: 62-71.

[22] The Snort Intrusion Detection System [Online]. 2020 [cit. 2020-10-21]. Dostupné z: https://www.snort.org/

[23] ALAZAB, Ammar, et al. Using response action with intelligent intrusion detection and prevention system against web application malware. Information Management & Computer Security, 2014.

[24] CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly detection: A survey. ACM computing surveys (CSUR), 2009, 41.3: 1-58.

[25] AHMED, Mohiuddin; MAHMOOD, Abdun Naser; HU, Jiankun. A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 2016, 60: 19-31.

[26] MIRSKY, Yisroel, et al. Anomaly detection for smartphone data streams. Pervasive and Mobile Computing, 2017, 35: 83-107.

[27] FANG, Fei, et al. (ed.). Artificial intelligence and conservation. Cambridge University Press, 2019.

[28] GOLDSTEIN, Markus; UCHIDA, Seiichi. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one, 2016, 11.4: e0152173.

[29] EBRAHIMI, Mohammadreza, et al. Recognizing predatory chat documents using semi-supervised anomaly detection. Electronic Imaging, 2016, 2016.17: 1-9.

[30] HUTTER, Frank; KOTTHOFF, Lars; VANSCHOREN, Joaquin. Automated machine learning: methods, systems, challenges. Springer Nature, 2019.

[31] NIU, Xuetong; WANG, Li; YANG, Xulei. A comparison study of credit card fraud detection: Supervised versus unsupervised. arXiv preprint arXiv:1904.10604, 2019.

[32] RAJENDRAN, Sreeraj, et al. Crowdsourced wireless spectrum anomaly detection. IEEE Transactions on Cognitive Communications and Networking, 2019, 6.2: 694-703.

[33] RUFF, Lukas, et al. Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906.02694, 2019.

[34] CAMACHO, Jose, et al. Semi-supervised multivariate statistical network monitoring for learning security threats. IEEE Transactions on Information Forensics and Security, 2019, 14.8: 2179-2189.

[35] ARUNRAJ, Nari S., et al. Comparison of supervised, semi-supervised and unsupervised learning methods in network intrusion detection system (NIDS) application. Anwendungen und Konzepte der Wirtschaftsinformatik, 2017, 6.

[36] DIVYA, K. T.; KUMARAN, N. Senthil. Improved Outlier Detection Using Classic KNN Algorithm. 2016.

[37] SAARI, Lukas. Detecting Performance Anomalies in a Mobile Application with Unsupervised Machine Learning. 2019.

[38] SOKOLOV, Alexander N.; PYATNITSKY, Ilya A.; ALABUGIN, Sergei K. Research of classical machine learning methods and deep learning models effectiveness in detecting anomalies of industrial control system. In: 2018 Global Smart Industry Conference (GloSIC). IEEE, 2018. p. 1-6.

[39] LIU, Junjiao, et al. A novel intrusion detection algorithm for industrial control systems based on CNN and process state transition. In: 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC). IEEE, 2018. p. 1-8.

[40] KRAVCHIK, Moshe; SHABTAI, Asaf. Detecting cyber attacks in industrial control systems using convolutional neural networks. In: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. 2018. p. 72-83.

[41] MIAO, Jianyu; NIU, Lingfeng. A survey on feature selection. Procedia Computer Science, 2016, 91: 919-926.

[42] GOODFELLOW, et al. Deep learning. MIT Press. [online]. 2016 [cit. 2020-10-21]. Dostupné z: http://www.deeplearningbook.org

[43] LEMAY, Antoine; FERNANDEZ, José M. Providing {SCADA} network data sets for intrusion detection research. In: 9th Workshop on Cyber Security Experimentation and Test ({CSET} 16). 2016.

[44] GOH, Jonathan, et al. A dataset to support research in the design of secure water treatment systems. In: International Conference on Critical Information Infrastructures Security. Springer, Cham, 2016. p. 88-99.

[45] MORRIS, Thomas H.; THORNTON, Zach; TURNIPSEED, Ian. Industrial control system simulation and data logging for intrusion detection system research. 7th annual southeastern cyber security summit, 2015, 3-4.

[46] ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 1958, 65.6: 386.

[47] ARCOLANO, Nicholas; RUDOY, Daniel. One-class support vector machines: Methods and applications. Harvard University, Final Project Presentation, 2008, 32.

[48] LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008. p. 413-422.

[49] LIPTON, Zachary C.; BERKOWITZ, John; ELKAN, Charles. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019, 2015.

[50] HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. Neural computation, 1997, 9.8: 1735-1780.

[51] GRAVES, Alex; JAITLY, Navdeep; MOHAMED, Abdel-rahman. Hybrid speech recognition with deep bidirectional LSTM. In: 2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013. p. 273-278.

[52] KOCHENDERFER, Mykel J.; WHEELER, Tim A. Algorithms for optimization. Mit Press, 2019.

[53] HOLLAND, John Henry, et al. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[54] BERGSTRA, James S., et al. Algorithms for hyper-parameter optimization. In: Advances in neural information processing systems. 2011. p. 2546-2554.

[55] YU, Tong; ZHU, Hong. Hyper-Parameter Optimization: A Review of Algorithms and Applications. arXiv preprint arXiv:2003.05689, 2020.

[56] NGUYEN, Hoang-Phuong; LIU, Jie; ZIO, Enrico. A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. Applied Soft Computing, 2020, 89: 106116.

[57] BERGSTRA, James; YAMINS, Daniel; COX, David. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: International conference on machine learning. 2013. p. 115-123.

[58] TZENG, Gwo-Hshiung; HUANG, Jih-Jeng. Multiple attribute decision making: methods and applications. CRC press, 2011.

[59] VAVREK, Roman, et al. Effectiveness of Use of MCDM Methods in the Terms of Local Self-Government. In: Advances in Applied Economic Research. Springer, Cham, 2017. p. 279-288.

[60] NISBET, Robert; ELDER, John; MINER, Gary. Handbook of statistical analysis and data mining applications. Academic Press, 2009.

[61] LI, Ning; SHEPPERD, Martin; GUO, Yuchen. A systematic review of unsupervised learning techniques for software defect prediction. Information and Software Technology, 2020, 106287.

[62] VIJAY, Kotu; BALA, Deshpande. Data Science (Second Edition), Chapter 8 - Model Evaluation, 2019. p. 263-279, ISBN 9780128147610.

[63] RIBEIRO, Marco Tulio; SINGH, Sameer; GUESTRIN, Carlos. " Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016. p. 1135-1144.

[64] DEMŠAR, Janez. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research, 2006, 7.Jan: 1-30.

10. SEZNAM OBRÁZKŮ

11. SEZNAM TABULEK

12. SEZNAM POUŽITÝCH ZKRATEK

ack	Acknowledgment
ANN	Artificial Neural Network
APT	Advanced Persistent Threat
CA	Cyber Attack
CERT	Computer Emergency Response Team
CIS	Center for Internet Security
CRC	Cyclic Redundancy Check
csv	Comma-Separated Values
CVSS	Common Vulnerability Scoring System
cwr	Congestion Window Reduced
ČR	Česká republika
DCS	Distributed Control System
DNP3	Distributed Network Protocol 3
DoS	Denial of Service
DT	Decision Tree
ENISA	European Union Agency for Network and Information
EU	Evropské unie
FF	Fitness Function
fin	Last packet
FIRST	Forum of Incident Response and Security Teams
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GA	Genetický algoritmus
GS	Grid Search
HCl	Kyselina chlorovodíková
HMI	Human machine interface
I/O	Input/output
ICS	Industrial control system
ICS-CERT	Industrial control system – Computer Emergency Response Team
ICT	Information and communication technologies
IDS	Intrusion Detection System
IF	Random Forest
IIoT	Industrial Internet of Things

IoT	Internet of Thinks
IP	Internet protocol
IPS	Intrusion Prevention System
ISACA	Information Systems Audit and Control Association
ISC	International Information Systems Security Certification Consortium
ISSA	Information Systems Security Association
IT	Informační technologie
ITIL	Information Technology Infrastructure Library
KI	Kritická infrastruktura
LRC	Longitudinal Redundancy Check
LSTM	Long Short Term Memory
Mbtcp	Modbus Transmission Control Protocol
MCA	Multicriterial Analysis
MCC	Matthews Correlation Coefficient
MH	Multikriteriálního Hodnocení
MSE	Mean Squared Error
MSS	The maximum segment size
MTU	Master Terminal Unit
NaOCl	Chlornan sodný
NIST	The National Institute of Standards
OCSVM	One-class Support Vector Machines
OF	Objective Function
OHE	One-Hot Encoder
PCA	Principal Component Analysis
pcap	Packet capturing
PDU	Protocol Data Unit
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
PPV	Positive Predictive Value
Prec	Precision
PSH	Push flag
RF	Random Forest
ROC	Receiver Operating Characteristic
RS	Grid search
RTU	Remote Terminal Unit
SANS	Escal Institute of Advanced Technologies

SCADA	Supervisory Control And Data Acquisition
SI	Mezinárodní systém jednotek
SIDS	Signature Intrusion Detection System
SMBO	Sequential model-based optimization
SVM	Support vector machines
SVM	Support Vector Machine
syn	Synchronize sequence numbers
Tcp	Transmission Control Protocol
TN	True Negative
TOPSIS	Technique for Order of Preference by Similarity to Ideal
TP	True Positive
TPE	Tree-structured Parzen Estimator
TPR	True Positive Rate
USA	United States of America

13. PŘÍLOHY

Příloha A: Experimenty pro nalezení vhodné kombinace pro úpravu datasetu.

Příloha B: Porovnání algoritmů strojového učení při základním nastavení.

Příloha C: Výsledky algoritmu OCSVM pro rozdílné hodnoty gamma parametru v rámci tří datasetů.

Příloha D: Optimalizace vybraných algoritmů strojového učení.

Příloha E: Výsledky algoritmů neuronová síť, LSTM a IF pro finální nastavení hyperparametrů v rámci tří datasetů.

Příloha F: Porovnání metriky M_{FPR} *pro jednotlivá řešení.*

Příloha G: Ověření v*ýsledků algoritmů neuronová síť, LSTM a IF pro finální nastavení hyperparametrů v rámci tří datasetů.*

Příloha H: Porovnání metriky M_{FPR} *pro jednotlivá řešení vzhledem k ověření systému detekce anomálií.*

Příloha A: Experimenty pro nalezení vhodné kombinace pro úpravu datasetu.

Neuronová síť – souhrnné výsledky pro různé techniky pro úpravu datasetů

Finální výsledky pro 900 modelů jsou uvedeny v Tab. 31 a Tab. 32 pro jednotlivé kybernetické útoky a techniky pro úpravu dat. Pro každou kombinaci technik pro úpravu datasetu jsou vypočteny následující veličiny: maximální hodnota, minimální hodnota a průměr z vybraných hodnotících metrik.

Tab. 31 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro první dva kybernetické útoky – neuronová síť (dataset 1). [vlastní zdroj]

		CA1_1					CA1_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Aritmetický průměr; normalizace <0,1>	**Průměr**	0.672	0.651	0.752	0.019	0.010	0.823	0.619	0.926	0.111	0.013
	Max	0.971	0.968	0.991	0.069	0.016	0.990	0.971	0.997	0.508	0.020
	Min	0.157	0.084	0.167	0.001	0.008	0.587	0.188	0.716	0.005	0.011
Aritmetický průměr; normalizace <-1,1>	**Průměr**	0.665	0.646	0.761	0.018	0.010	0.761	0.450	0.856	0.222	0.013
	Max	0.804	0.800	0.965	0.071	0.014	0.825	0.656	0.978	0.520	0.017
	Min	0.164	0.090	0.171	0.002	0.008	0.248	0.072	0.688	0.029	0.011
Aritmetický průměr; standardizace	**Průměr**	0.732	0.713	0.804	0.016	0.010	0.755	0.379	0.804	0.321	0.013
	Max	0.783	0.775	0.921	0.041	0.013	0.786	0.495	0.872	0.503	0.017
	Min	0.517	0.474	0.530	0.005	0.008	0.710	0.198	0.719	0.191	0.011
Medián; normalizace <0,1>	**Průměr**	0.722	0.703	0.795	0.017	0.010	0.784	0.500	0.876	0.194	0.013
	Max	0.966	0.963	1.000	0.044	0.014	0.984	0.954	0.993	0.508	0.021
	Min	0.343	0.292	0.389	0.000	0.009	0.702	0.181	0.713	0.009	0.011
Medián; normalizace <-1,1>	**Průměr**	0.680	0.667	0.812	0.014	0.010	0.768	0.464	0.864	0.210	0.013
	Max	0.804	0.800	0.962	0.060	0.013	0.809	0.605	0.954	0.509	0.017
	Min	0.171	0.125	0.242	0.002	0.008	0.702	0.181	0.713	0.061	0.010
Medián; standardizace	**Průměr**	0.740	0.723	0.821	0.014	0.011	0.766	0.419	0.825	0.280	0.013
	Max	0.785	0.779	0.932	0.041	0.020	0.799	0.526	0.880	0.479	0.020
	Min	0.556	0.515	0.558	0.005	0.008	0.711	0.214	0.726	0.177	0.011
Náhrada konstantou; normalizace <0,1>	**Průměr**	0.724	0.705	0.793	0.017	0.011	0.790	0.507	0.874	0.200	0.013
	Max	0.975	0.973	1.000	0.069	0.014	0.997	0.992	0.997	0.503	0.024
	Min	0.173	0.101	0.182	0.000	0.008	0.706	0.198	0.719	0.005	0.011
Náhrada konstantou; normalizace <-1,1>	**Průměr**	0.467	0.435	0.560	0.033	0.011	0.770	0.481	0.875	0.192	0.013
	Max	0.800	0.800	0.976	0.077	0.014	0.833	0.647	0.963	0.518	0.017
	Min	0.115	0.044	0.127	0.002	0.008	0.279	0.120	0.710	0.049	0.011
Náhrada konstantou; standardizace	**Průměr**	0.618	0.600	0.744	0.019	0.010	0.764	0.425	0.833	0.264	0.012
	Max	0.789	0.780	0.915	0.042	0.014	0.795	0.532	0.892	0.483	0.015
	Min	0.448	0.403	0.481	0.005	0.008	0.710	0.214	0.727	0.155	0.011

Tab. 32 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro třetí a čtvrtý kybernetický útok – neuronová síť (dataset 1). [vlastní zdroj]

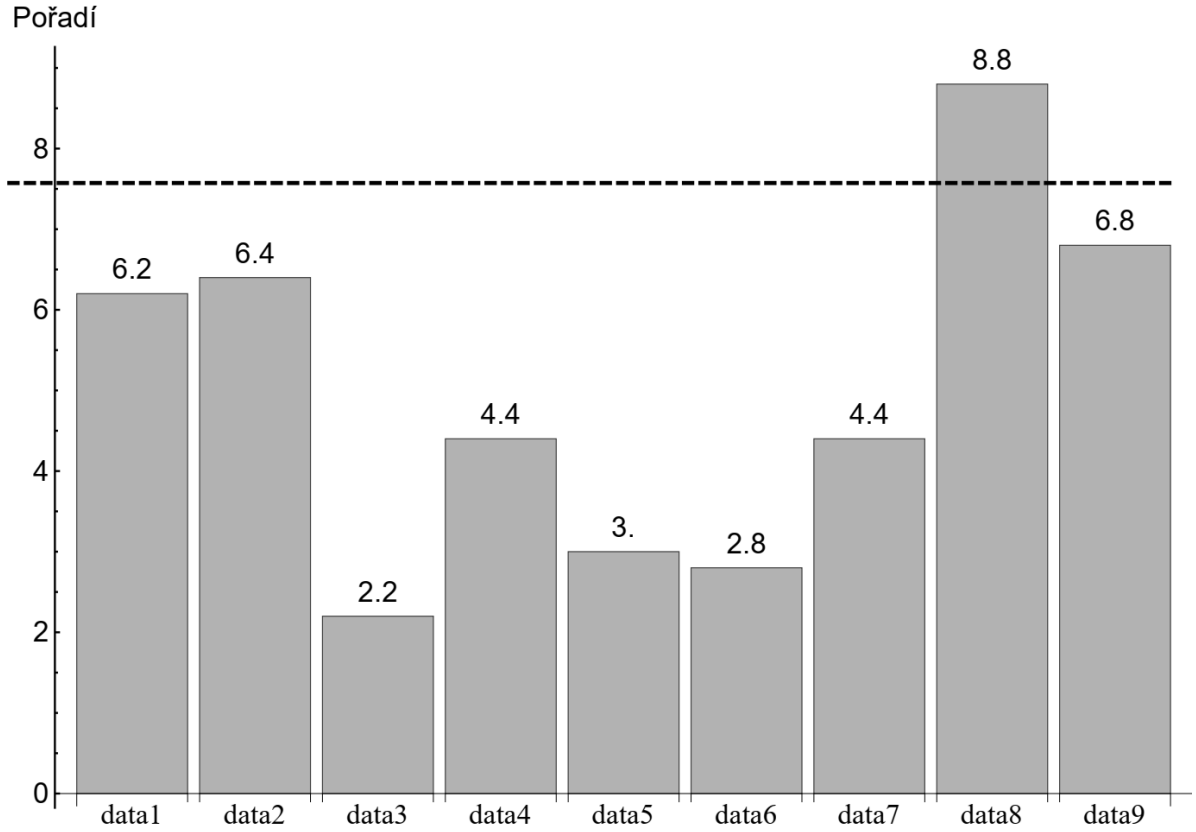
		CA1_3					CA1_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Aritmetický průměr; normalizace <0,1>	**Průměr**	0.590	0.587	0.678	0.007	0.041	0.053	0.056	0.087	0.008	0.063
	Max	0.905	0.907	1.000	0.061	0.053	0.333	0.447	1.000	0.143	0.071
	Min	0.000	0.038	0.000	0.000	0.038	0.000	-0.012	0.000	0.000	0.059
Aritmetický průměr; normalizace <-1,1>	**Průměr**	0.593	0.587	0.641	0.007	0.042	0.020	0.020	0.036	0.012	0.065
	Max	0.797	0.796	0.873	0.026	0.054	0.308	0.365	0.667	0.129	0.122
	Min	0.000	-0.025	0.000	0.002	0.038	0.000	-0.012	0.000	0.000	0.058
Aritmetický průměr; standardizace	**Průměr**	0.276	0.260	0.289	0.015	0.043	0.000	-0.007	0.000	0.058	0.066
	Max	0.556	0.546	0.580	0.039	0.057	0.000	-0.006	0.000	0.110	0.092
	Min	0.000	-0.030	0.000	0.009	0.038	0.000	-0.010	0.000	0.038	0.059
Medián; normalizace <0,1>	**Průměr**	0.554	0.551	0.638	0.006	0.044	0.099	0.108	0.166	0.003	0.068
	Max	0.812	0.811	0.918	0.020	0.057	0.471	0.478	1.000	0.072	0.097
	Min	0.000	-0.016	0.000	0.001	0.039	0.000	-0.008	0.000	0.000	0.061
Medián; normalizace <-1,1>	**Průměr**	0.414	0.405	0.462	0.010	0.041	0.048	0.049	0.070	0.005	0.064
	Max	0.764	0.759	0.815	0.020	0.076	0.222	0.223	0.500	0.122	0.086
	Min	0.028	0.007	0.030	0.004	0.038	0.000	-0.011	0.000	0.000	0.058
Medián; standardizace	**Průměr**	0.259	0.243	0.270	0.016	0.045	0.000	-0.008	0.000	0.060	0.068
	Max	0.662	0.655	0.671	0.040	0.065	0.000	-0.006	0.000	0.087	0.080
	Min	0.000	-0.031	0.000	0.007	0.038	0.000	-0.009	0.000	0.040	0.059
Náhrada konstantou; normalizace <0,1>	**Průměr**	0.550	0.545	0.622	0.008	0.046	0.106	0.116	0.188	0.008	0.069
	Max	0.881	0.880	0.926	0.074	0.055	0.625	0.645	1.000	0.110	0.080
	Min	0.000	-0.042	0.000	0.002	0.040	0.000	-0.011	0.000	0.000	0.064
Náhrada konstantou; normalizace <-1,1>	**Průměr**	0.517	0.509	0.563	0.008	0.043	0.023	0.023	0.037	0.007	0.067
	Max	0.759	0.754	0.825	0.028	0.054	0.286	0.316	0.500	0.062	0.080
	Min	0.000	-0.026	0.000	0.003	0.038	0.000	-0.008	0.000	0.000	0.060
Náhrada konstantou; standardizace	**Průměr**	0.178	0.160	0.186	0.017	0.040	0.000	-0.007	0.000	0.059	0.061
	Max	0.514	0.504	0.536	0.022	0.046	0.000	-0.006	0.000	0.083	0.069
	Min	0.014	-0.008	0.014	0.010	0.038	0.000	0.009	0.000	0.044	0.058

V následujících odstavcích jsou uvedeny výsledky pro p-hodnotu a Friedmanův test včetně Nemenyiho kritické vzdálenosti.

Tab. 33 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	7.1524163568	0.0000211834

Podle Tab. 33 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



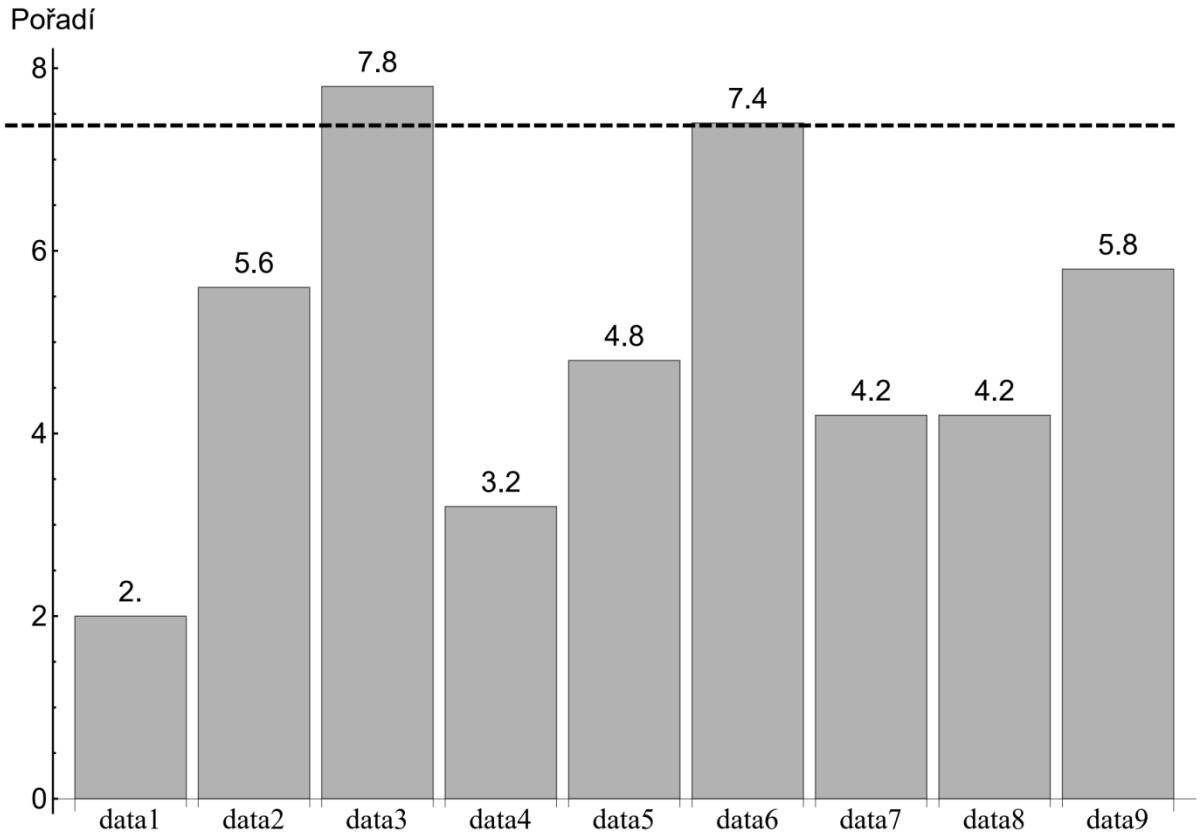
Obr. 47: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku - CA1_1. [vlastní zdroj]

V rámci Obr. 47 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data8 oproti ostatním.

Tab. 34 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	3.5376884422	0.0048578011

Podle Tab. 34 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



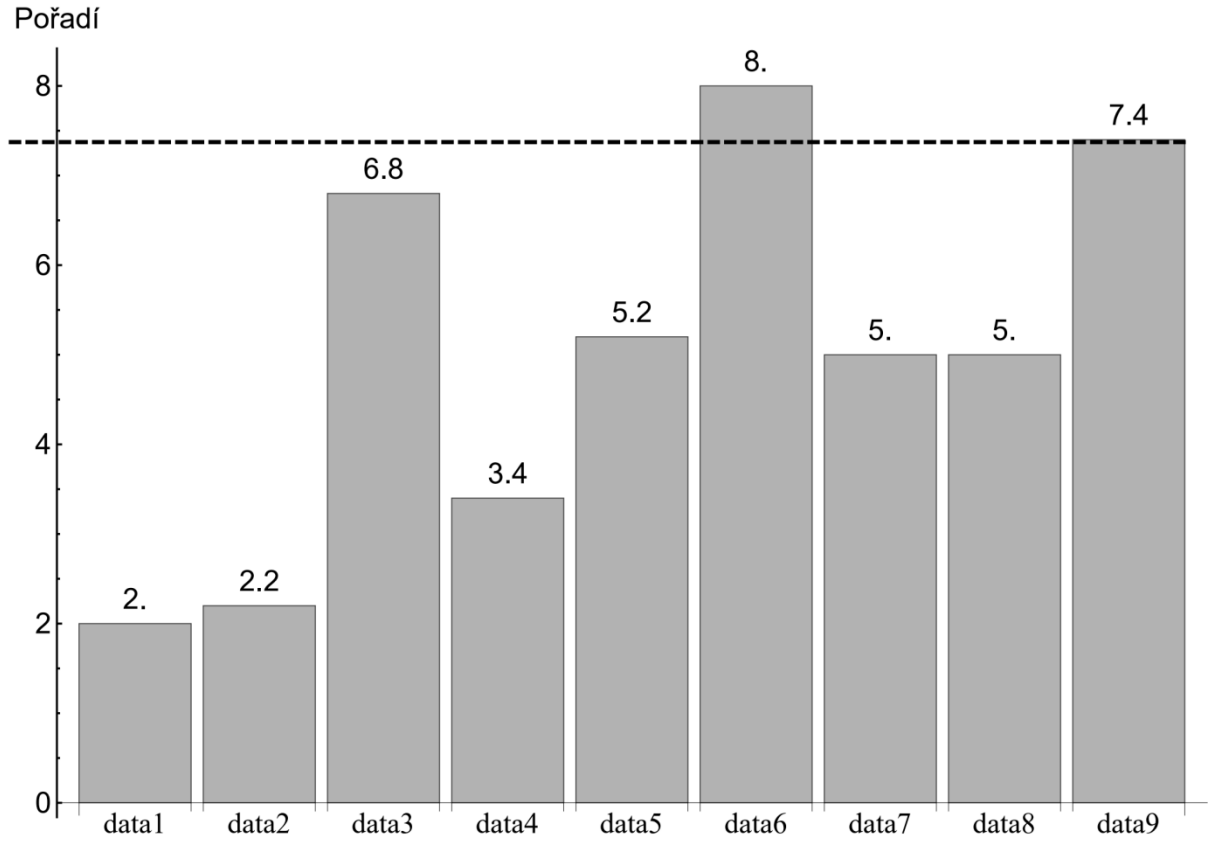
Obr. 48: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

V rámci Obr. 48 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data3 a data6 oproti ostatním.

Tab. 35 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	6.6382978723	0.0000418449

Podle Tab. 35 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



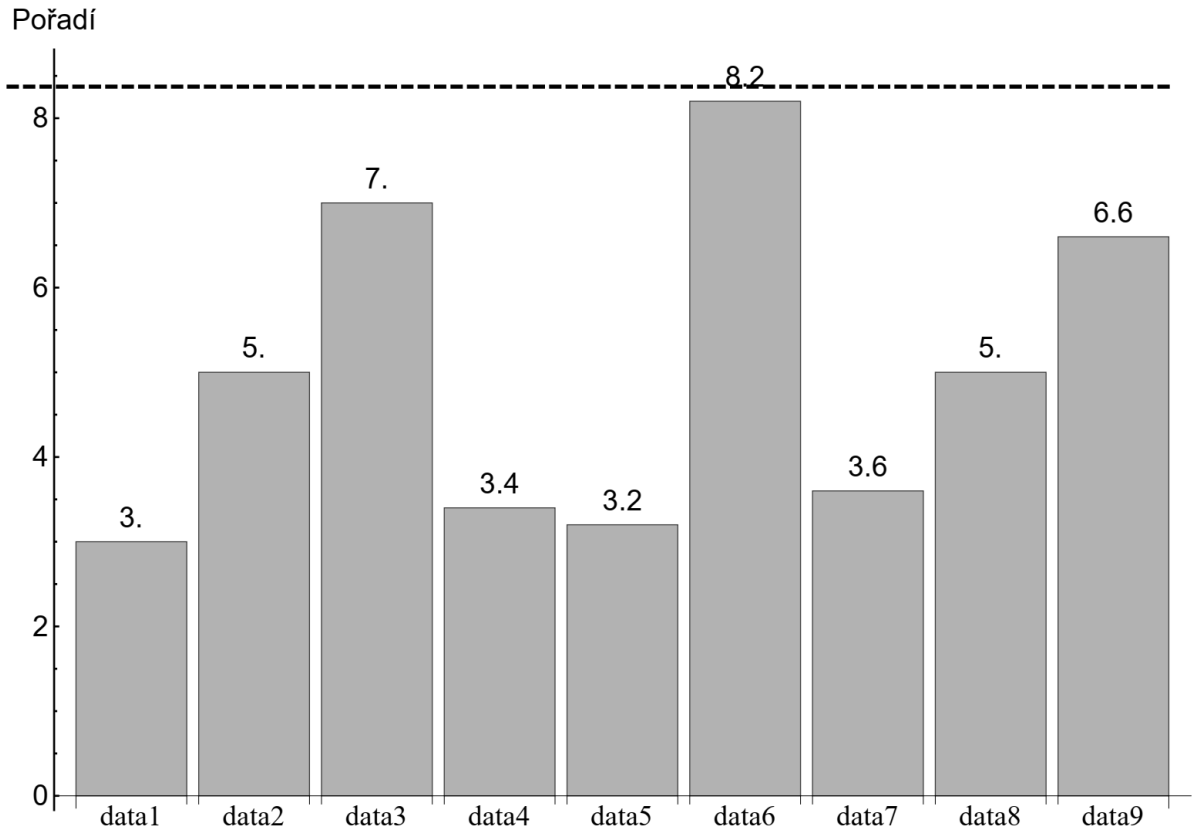
Obr. 49: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

V rámci Obr. 49 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data6 a data9 oproti ostatním.

Tab. 36 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	3.72845953	0.0034996611

Podle Tab. 36 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 50: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

V rámci Obr. 50 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

LSTM – souhrnné výsledky pro různé techniky pro úpravu datasetů

Finální výsledky pro 900 modelů jsou uvedeny v Tab. 37 a Tab. 38 pro jednotlivé kybernetické útoky a techniky pro úpravu dat. Pro každou kombinaci technik pro úpravu datasetu jsou vypočteny následující parametry: maximální hodnota, minimální hodnota a průměr z vybraných hodnotících metrik.

Tab. 37 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro první dva kybernetické útoky – LSTM (dataset 1). [vlastní zdroj]

		CA1_1					CA1_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Aritmetický průměr; normalizace <0,1>	**Průměr**	0.885	0.875	0.911	0.008	0.011	0.958	0.899	0.982	0.031	0.021
	Max	0.970	0.968	0.991	0.091	0.012	0.997	0.993	0.999	0.875	0.023
	Min	0.017	-0.075	0.017	0.001	0.009	0.493	-0.338	0.548	0.002	0.016

Aritmetický průměr; normalizace <-1,1>	**Průměr**	0.732	0.709	0.771	0.021	0.011	0.908	0.749	0.933	0.126	0.021
	Max	0.944	0.940	0.982	0.133	0.014	0.984	0.951	0.991	0.882	0.024
	Min	0.000	-0.114	0.000	0.002	0.009	0.544	-0.340	0.549	0.018	0.017
Aritmetický průměr; standardizace	**Průměr**	0.415	0.361	0.423	0.053	0.011	0.764	0.299	0.767	0.460	0.021
	Max	0.598	0.560	0.609	0.159	0.014	0.925	0.777	0.926	0.804	0.027
	Min	0.000	-0.127	0.000	0.033	0.009	0.574	-0.235	0.584	0.148	0.017
Medián; normalizace <0,1>	**Průměr**	0.867	0.855	0.896	0.010	0.011	0.954	0.886	0.977	0.039	0.021
	Max	0.966	0.963	0.983	0.151	0.014	0.999	0.996	0.999	0.907	0.023
	Min	0.000	-0.123	0.000	0.002	0.009	0.415	-0.491	0.462	0.002	0.017
Medián; normalizace <-1,1>	**Průměr**	0.713	0.687	0.737	0.025	0.011	0.917	0.765	0.933	0.127	0.022
	Max	0.958	0.954	0.973	0.152	0.014	0.990	0.971	0.992	0.870	0.026
	Min	0.000	-0.123	0.000	0.002	0.010	0.550	-0.323	0.555	0.017	0.017
Medián; standardizace	**Průměr**	0.436	0.384	0.444	0.051	0.011	0.751	0.262	0.755	0.484	0.021
	Max	0.703	0.676	0.722	0.084	0.014	0.940	0.821	0.941	0.845	0.023
	Min	0.052	-0.034	0.054	0.025	0.009	0.534	-0.319	0.550	0.118	0.017
Náhrada konstantou; normalizace <0,1>	**Průměr**	0.818	0.807	0.887	0.010	0.011	0.940	0.843	0.968	0.058	0.021
	Max	0.970	0.968	1.000	0.091	0.013	0.996	0.989	0.999	0.997	0.023
	Min	0.009	-0.081	0.009	0.000	0.010	0.483	-0.513	0.489	0.002	0.017
Náhrada konstantou; normalizace <-1,1>	**Průměr**	0.183	0.166	0.330	0.024	0.012	0.798	0.481	0.854	0.256	0.021
	Max	0.706	0.679	0.865	0.089	0.013	0.929	0.789	0.938	0.880	0.024
	Min	0.000	-0.074	0.000	0.004	0.010	0.552	-0.330	0.554	0.116	0.017
Náhrada konstantou; standardizace	**Průměr**	0.421	0.367	0.425	0.053	0.012	0.732	0.209	0.737	0.516	0.021
	Max	0.631	0.596	0.633	0.083	0.014	0.910	0.730	0.911	0.864	0.026
	Min	0.101	0.017	0.103	0.034	0.009	0.507	-0.364	0.529	0.178	0.017

Tab. 38 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro třetí a čtvrtý kybernetický útok – LSTM (dataset 1). [vlastní zdroj]

		CA1_3					**CA1_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Aritmetický průměr; normalizace <0,1>	**Průměr**	0.722	0.724	0.825	0.003	0.350	0.276	0.280	0.336	0.004	0.069
	Max	0.966	0.965	1.000	0.036	0.417	0.824	0.837	1.000	0.248	0.093
	Min	0.000	-0.029	0.000	0.000	0.336	0.000	-0.017	0.000	0.000	0.062
Aritmetický průměr; normalizace <-1,1>	**Průměr**	0.415	0.404	0.461	0.016	0.374	0.071	0.073	0.107	0.025	0.073
	Max	0.775	0.771	0.860	0.134	0.489	0.375	0.387	1.000	0.286	0.094
	Min	0.000	-0.059	0.000	0.002	0.344	0.000	-0.019	0.000	0.000	0.063
Aritmetický průměr; standardizace	**Průměr**	0.060	0.039	0.062	0.021	0.390	0.000	-0.013	0.000	0.155	0.073
	Max	0.414	0.401	0.429	0.022	0.594	0.000	-0.003	0.000	0.254	0.100
	Min	0.027	0.004	0.027	0.012	0.341	0.000	-0.018	0.000	0.008	0.064
Medián; normalizace <0,1>	**Průměr**	0.715	0.715	0.812	0.006	0.354	0.225	0.230	0.283	0.011	0.070
	Max	0.959	0.958	0.986	0.202	0.673	0.778	0.782	1.000	0.434	0.090
	Min	0.000	-0.076	0.000	0.000	0.337	0.000	-0.026	0.000	0.000	0.063
Medián; normalizace <-1,1>	**Průměr**	0.493	0.482	0.541	0.016	0.372	0.045	0.042	0.055	0.040	0.074
	Max	0.800	0.798	0.864	0.160	0.550	0.375	0.387	0.500	0.372	0.085
	Min	0.000	-0.066	0.000	0.002	0.341	0.000	-0.023	0.000	0.000	0.065
Medián;	**Průměr**	0.044	0.022	0.045	0.022	0.353	0.000	-0.013	0.000	0.166	0.071

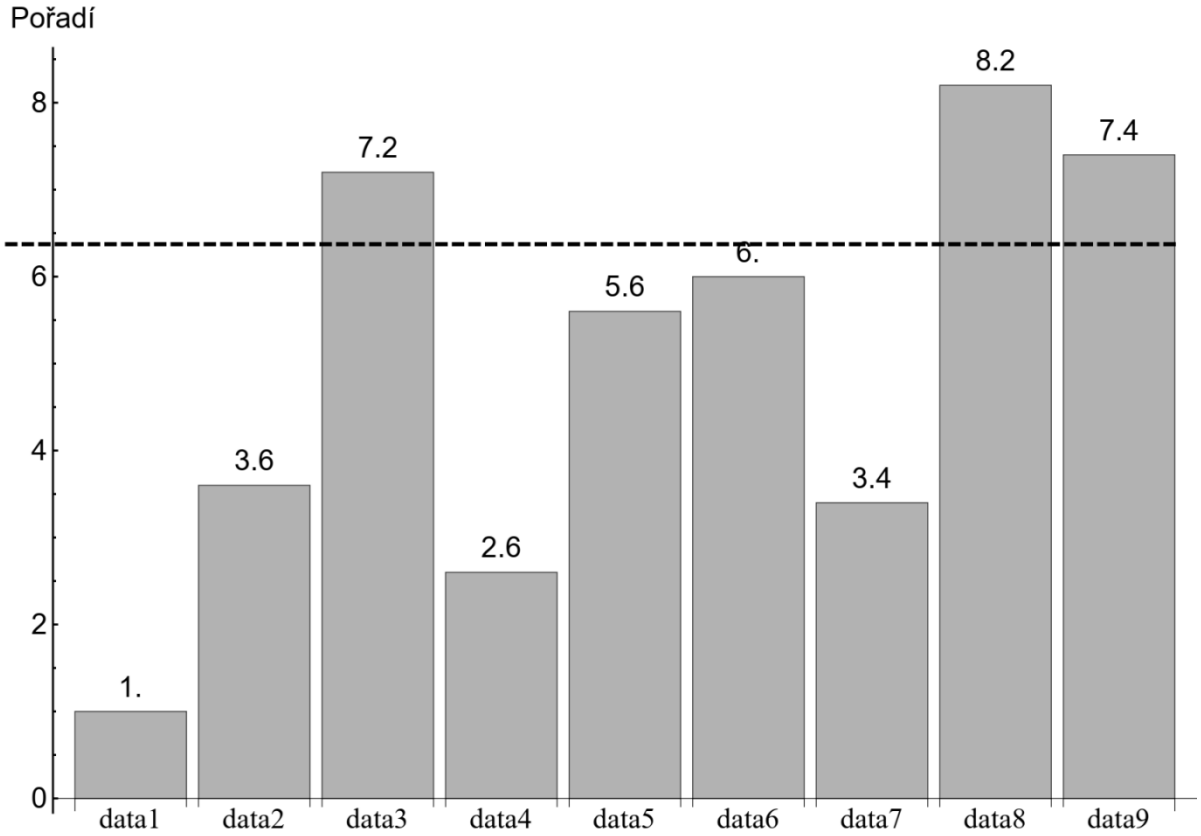
standardizace	**Max**	0.298	0.283	0.318	0.022	0.393	0.000	-0.004	0.000	0.252	0.081
	Min	0.014	-0.009	0.014	0.014	0.340	0.000	-0.017	0.000	0.016	0.063
Náhrada konstantou; normalizace <0,1>	**Průměr**	0.627	0.629	0.749	0.005	0.351	0.219	0.220	0.257	0.011	0.072
	Max	0.937	0.937	1.000	0.086	0.426	0.778	0.782	1.000	0.355	0.080
	Min	0.000	-0.046	0.000	0.000	0.337	0.000	-0.022	0.000	0.000	0.064
Náhrada konstantou; normalizace <-1,1>	**Průměr**	0.145	0.127	0.159	0.017	0.345	0.045	0.048	0.078	0.017	0.072
	Max	0.695	0.690	0.742	0.054	0.373	0.167	0.223	0.500	0.202	0.082
	Min	0.000	-0.036	0.000	0.005	0.337	0.000	-0.015	0.000	0.000	0.066
Náhrada konstantou; standardizace	**Průměr**	0.081	0.060	0.082	0.021	0.353	0.000	-0.010	0.000	0.111	0.071
	Max	0.523	0.512	0.527	0.022	0.506	0.000	-0.003	0.000	0.252	0.085
	Min	0.041	0.019	0.041	0.011	0.336	0.000	-0.017	0.000	0.010	0.064

V následujících odstavcích jsou uvedeny výsledky pro p-hodnotu a Friedmanův test včetně Nemenyiho kritické vzdálenosti.

Tab. 39 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	16.8333	$1.8183796623 \times 10^{-9}$

Podle Tab. 39 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



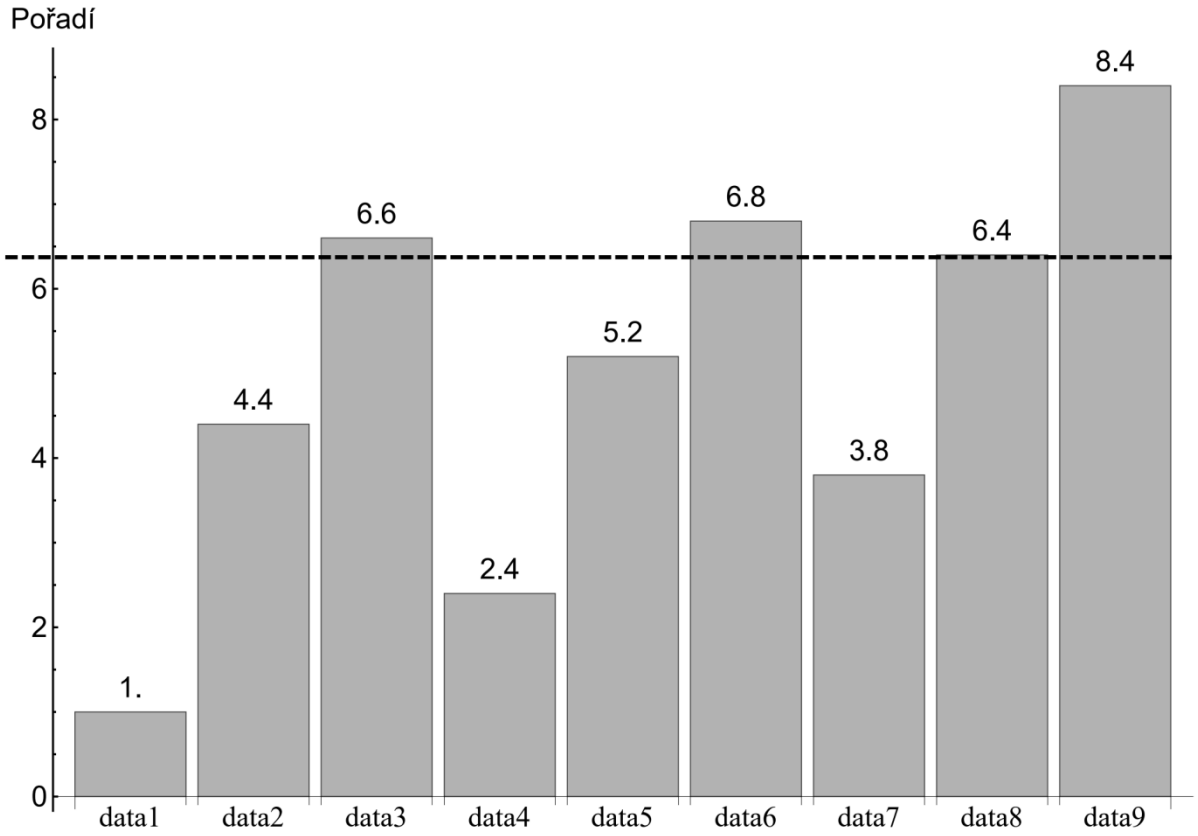
Obr. 51: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

V rámci Obr. 51 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data3, data8 a data9 oproti ostatním.

Tab. 40 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	10.9253731343	$2.8727346864 \times 10^{-7}$

Podle Tab. 40 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 52: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

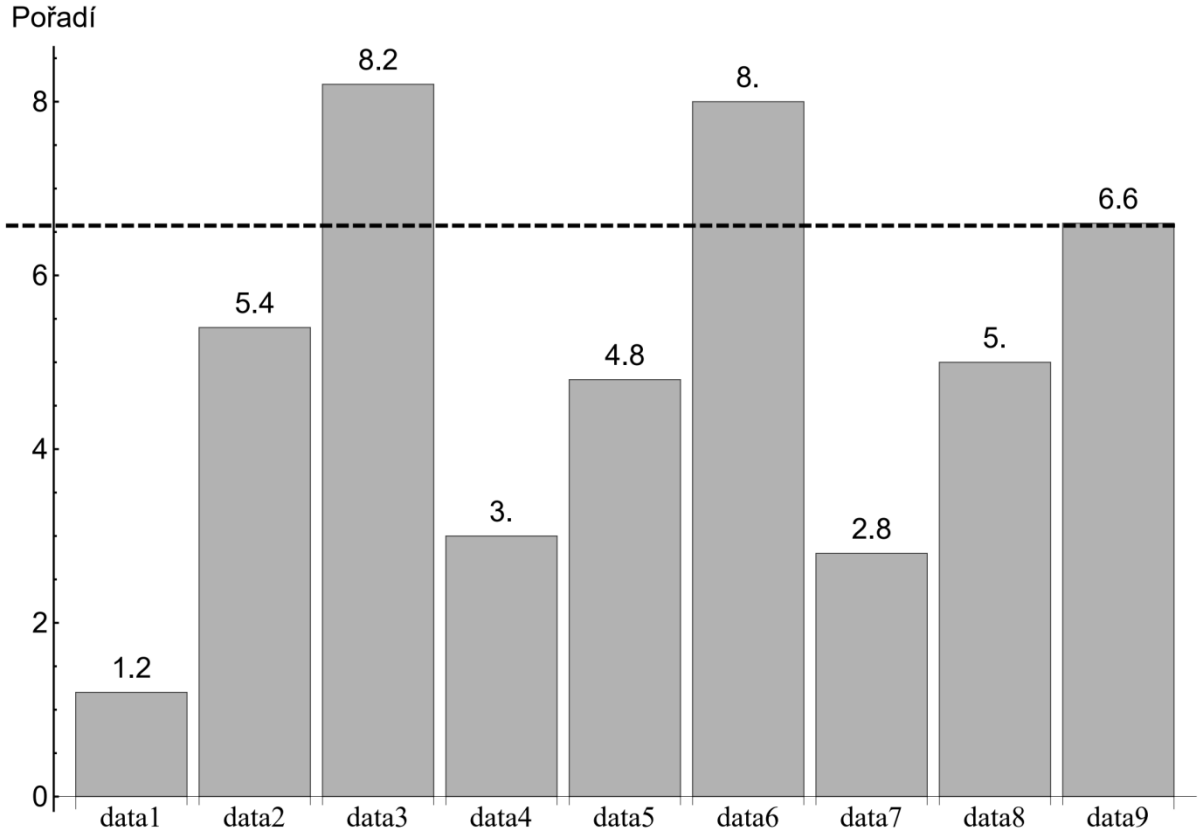
V rámci Obr. 52 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky

poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data3, data6, data8 a data9 oproti ostatním.

Tab. 41 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	12.3043478261	$7.5995072821 \times 10^{-8}$

Podle Tab. 41 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



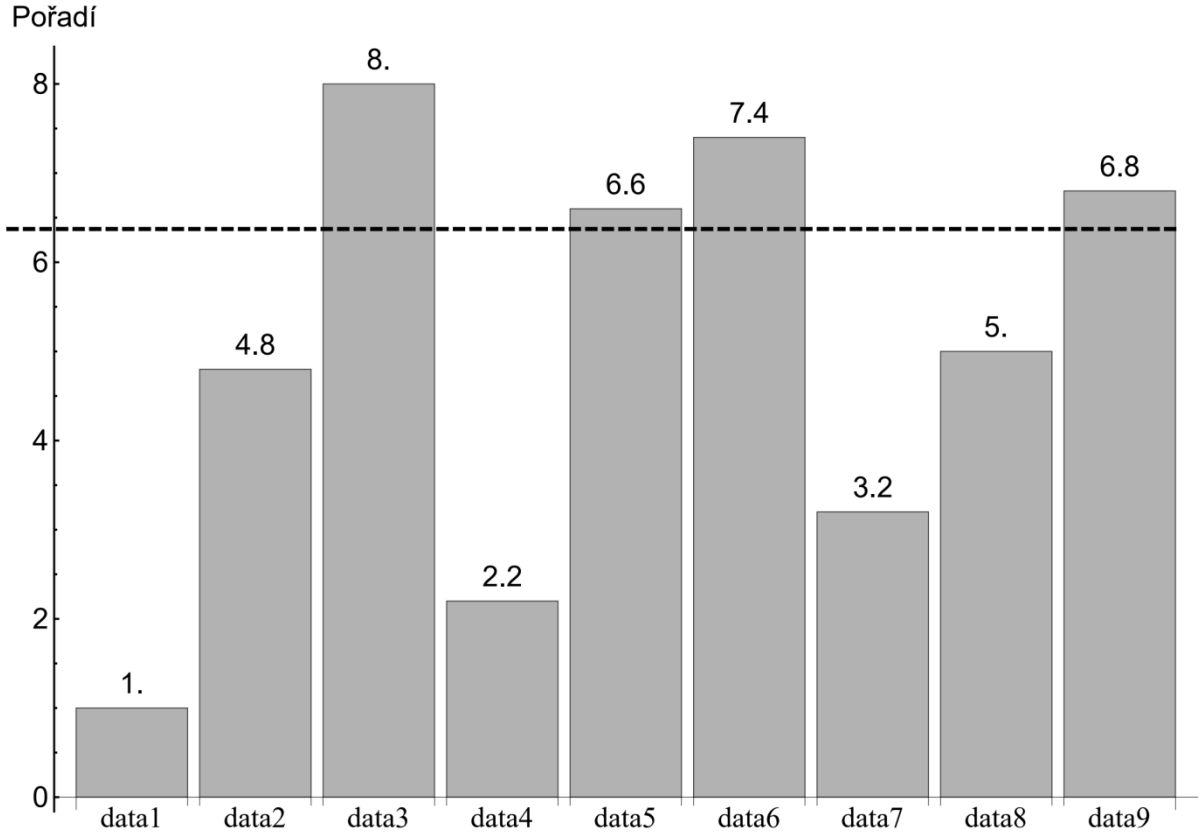
Obr. 53: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

V rámci Obr. 53 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data3, data6 a data9 oproti ostatním.

Tab. 42 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	16.55556	$2.2339607368 \times 10^{-9}$

Podle Tab. 42 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 54: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

V rámci Obr. 54 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data3, data5, data6 a data9 oproti ostatním.

Isolation Forest – souhrnné výsledky pro různé techniky pro úpravu datasetů

Finální výsledky pro 900 modelů pro jednotlivé kybernetické útoky a techniky pro úpravu dat jsou uvedeny v Tab. 43 a Tab. 44. Pro každou kombinaci technik

pro úpravu datasetu je vypočtena maximální hodnota, minimální hodnota a průměr z vybraných hodnotících metrik.

Tab. 43 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro první dva kybernetické útoky – Isolation Forest (dataset 1). [vlastní zdroj]

		CA1_1					CA1_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Aritmetický průměr; normalizace <0,1>	**Průměr**	0.307	0.242	0.223	0.158	0.068	0.620	0.417	0.866	0.086	0.077
	Max	0.526	0.518	0.388	0.202	0.077	0.805	0.610	0.966	0.119	0.094
	Min	0.027	-0.106	0.020	0.092	0.066	0.090	-0.056	0.546	0.044	0.074
Aritmetický průměr; normalizace <-1,1>	**Průměr**	0.393	0.340	0.297	0.131	0.068	0.753	0.533	0.936	0.078	0.077
	Max	0.500	0.478	0.406	0.184	0.099	0.797	0.599	0.970	0.113	0.086
	Min	0.302	0.230	0.221	0.080	0.066	0.698	0.450	0.908	0.034	0.075
Aritmetický průměr; standardizace	**Průměr**	0.391	0.337	0.304	0.132	0.067	0.743	0.549	0.964	0.041	0.076
	Max	0.582	0.542	0.543	0.227	0.074	0.798	0.629	0.996	0.082	0.083
	Min	0.270	0.199	0.180	0.049	0.065	0.714	0.496	0.932	0.005	0.074
Medián; normalizace <0,1>	**Průměr**	0.324	0.262	0.236	0.158	0.067	0.555	0.358	0.833	0.085	0.075
	Max	0.553	0.515	0.464	0.210	0.077	0.805	0.618	0.965	0.133	0.081
	Min	0.006	-0.119	0.005	0.074	0.065	0.056	-0.118	0.404	0.046	0.074
Medián; normalizace <-1,1>	**Průměr**	0.376	0.318	0.288	0.130	0.069	0.739	0.519	0.932	0.077	0.078
	Max	0.580	0.551	0.469	0.222	0.187	0.795	0.607	0.971	0.113	0.137
	Min	0.117	0.013	0.094	0.074	0.066	0.095	-0.095	0.481	0.035	0.074
Medián; standardizace	**Průměr**	0.384	0.328	0.299	0.135	0.070	0.752	0.556	0.959	0.047	0.080
	Max	0.595	0.566	0.590	0.248	0.109	0.811	0.642	0.996	0.085	0.100
	Min	0.222	0.132	0.148	0.037	0.066	0.206	0.103	0.798	0.005	0.074
Náhrada konstantou; normalizace <0,1>	**Průměr**	0.294	0.225	0.212	0.167	0.070	0.521	0.326	0.818	0.088	0.081
	Max	0.585	0.547	0.509	0.222	0.097	0.798	0.606	0.963	0.127	0.167
	Min	0.011	-0.122	0.008	0.061	0.066	0.039	-0.181	0.269	0.047	0.074
Náhrada konstantou; normalizace <-1,1>	**Průměr**	0.406	0.358	0.305	0.135	0.071	0.774	0.556	0.936	0.082	0.081
	Max	0.575	0.566	0.488	0.215	0.089	0.800	0.610	0.970	0.110	0.125
	Min	0.276	0.200	0.200	0.061	0.067	0.705	0.460	0.911	0.038	0.075
Náhrada konstantou; standardizace	**Průměr**	0.454	0.405	0.389	0.097	0.067	0.748	0.552	0.953	0.049	0.076
	Max	0.692	0.665	0.717	0.252	0.074	0.801	0.629	0.987	0.091	0.085
	Min	0.049	-0.056	0.042	0.025	0.065	0.093	-0.004	0.638	0.015	0.074

Tab. 44 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro třetí a čtvrtý kybernetický útok – Isolation Forest (dataset 1). [vlastní zdroj]

		CA1_3					CA1_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Aritmetický průměr; normalizace <0,1>	**Průměr**	0.171	0.177	0.108	0.082	0.119	0.001	-0.002	0.001	0.082	0.433
	Max	0.362	0.396	0.242	0.130	0.151	0.010	0.045	0.005	0.116	0.500
	Min	0.022	-0.019	0.014	0.027	0.114	0.000	-0.011	0.000	0.046	0.419
Aritmetický průměr; normalizace <-1,1>	**Průměr**	0.088	0.066	0.059	0.075	0.119	0.002	0.000	0.001	0.075	0.433
	Max	0.255	0.250	0.244	0.107	0.149	0.020	0.073	0.010	0.113	0.486

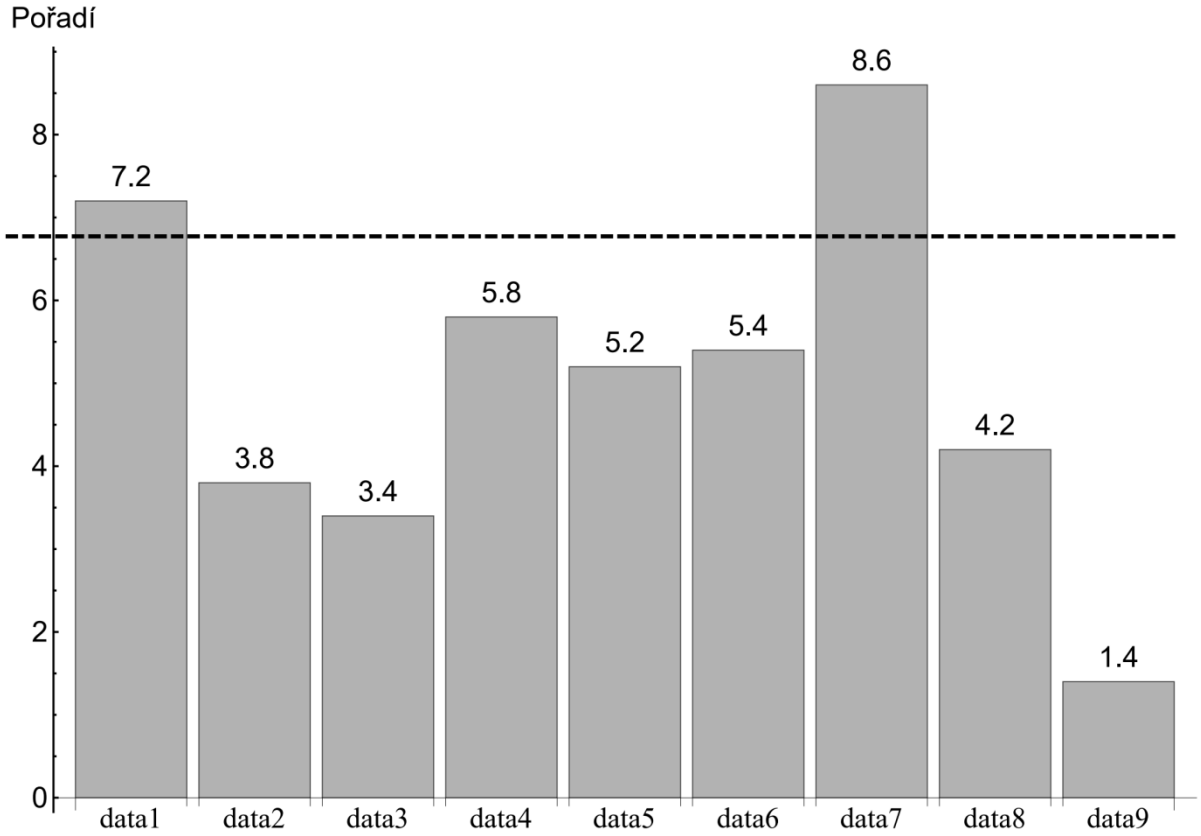
	Min	0.000	-0.050	0.000	0.019	0.116	0.000	-0.011	0.000	0.036	0.423
Aritmetický průměr; standardizace	**Průměr**	0.145	0.125	0.118	0.043	0.118	0.002	0.000	0.001	0.045	0.425
	Max	0.343	0.330	0.369	0.096	0.131	0.027	0.094	0.014	0.094	0.460
	Min	0.022	-0.017	0.014	0.010	0.115	0.000	-0.010	0.000	0.017	0.417
Medián; normalizace <0,1>	**Průměr**	0.181	0.186	0.116	0.079	0.119	0.001	-0.002	0.001	0.079	0.426
	Max	0.412	0.435	0.291	0.117	0.169	0.012	0.037	0.006	0.115	0.467
	Min	0.038	0.004	0.025	0.039	0.116	0.000	-0.011	0.000	0.030	0.416
Medián; normalizace <-1,1>	**Průměr**	0.064	0.036	0.044	0.072	0.122	0.001	-0.003	0.001	0.070	0.439
	Max	0.350	0.344	0.275	0.108	0.166	0.009	0.036	0.005	0.106	0.606
	Min	0.000	-0.050	0.000	0.012	0.117	0.000	-0.010	0.000	0.025	0.424
Medián; standardizace	**Průměr**	0.152	0.134	0.118	0.046	0.129	0.005	0.010	0.003	0.047	0.444
	Max	0.397	0.389	0.464	0.084	0.243	0.026	0.061	0.014	0.084	0.497
	Min	0.025	-0.007	0.018	0.009	0.118	0.000	-0.009	0.000	0.015	0.421
Náhrada konstantou; normalizace <0,1>	**Průměr**	0.175	0.184	0.109	0.086	0.128	0.001	-0.002	0.001	0.087	0.446
	Max	0.358	0.393	0.238	0.111	0.170	0.007	0.025	0.003	0.124	0.566
	Min	0.051	0.020	0.032	0.048	0.118	0.000	-0.011	0.000	0.042	0.419
Náhrada konstantou; normalizace <-1,1>	**Průměr**	0.167	0.163	0.113	0.070	0.129	0.001	-0.005	0.000	0.076	0.453
	Max	0.584	0.575	0.547	0.116	0.239	0.008	0.027	0.004	0.104	0.603
	Min	0.012	-0.031	0.007	0.012	0.119	0.000	-0.010	0.000	0.032	0.429
Náhrada konstantou; standardizace	**Průměr**	0.183	0.171	0.137	0.044	0.121	0.002	-0.001	0.001	0.045	0.429
	Max	0.443	0.430	0.422	0.074	0.140	0.013	0.046	0.007	0.087	0.472
	Min	0.061	0.033	0.043	0.015	0.117	0.000	-0.009	0.000	0.017	0.418

V následujících odstavcích jsou uvedeny výsledky pro p-hodnotu a Friedmanův test včetně Nemenyiho kritické vzdálenosti.

Tab. 45 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	6.10101	0.0000877982

Podle Tab. 45 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



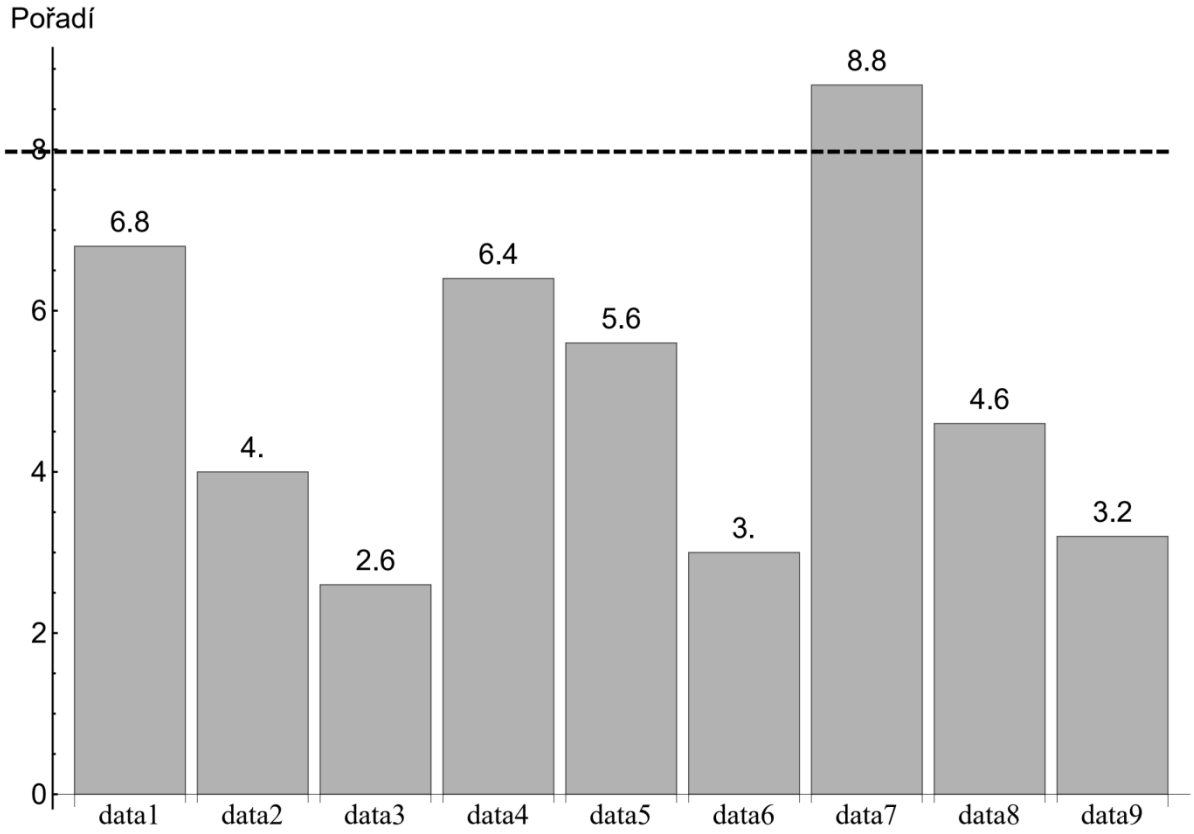
Obr. 55: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

V Obr. 55 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data1, data7 oproti ostatním.

Tab. 46 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	5.2879256966	0.0002867797

Podle Tab. 46 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



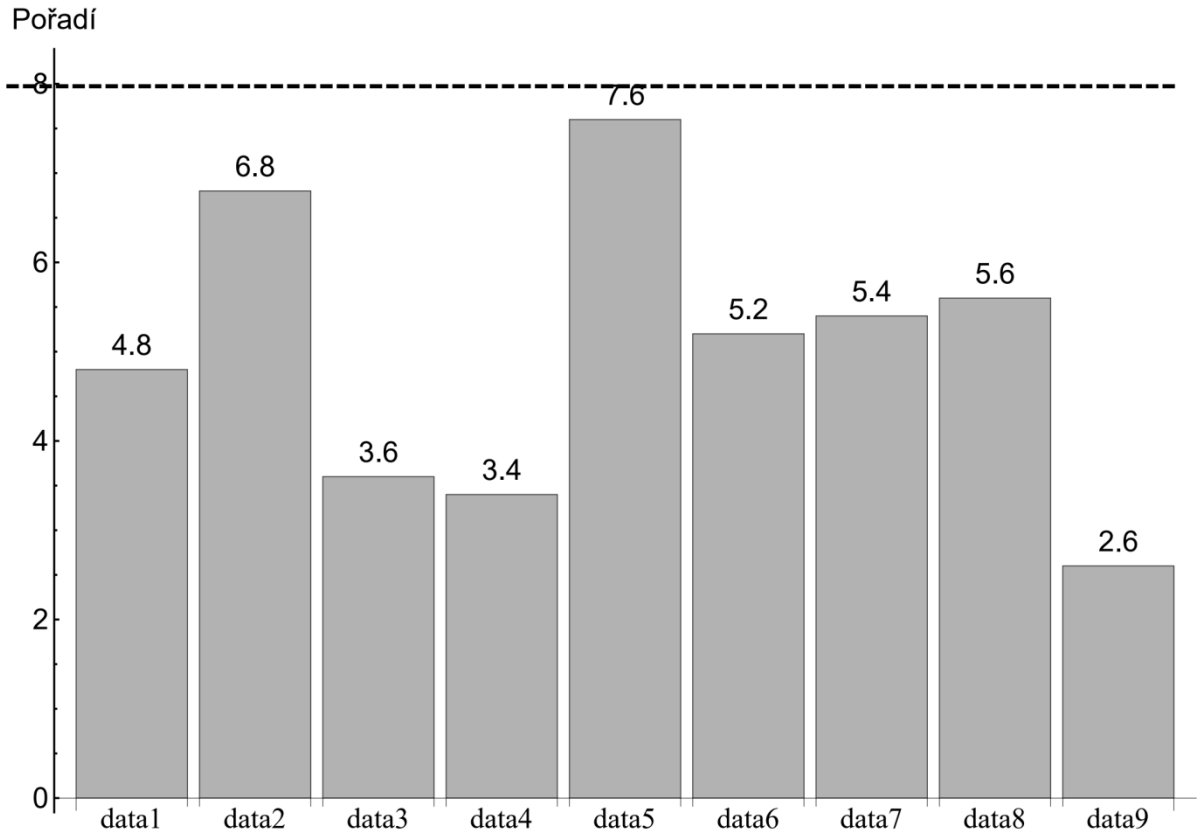
Obr. 56: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

V Obr. 56 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data7 oproti ostatním.

Tab. 47 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.1349693252	0.0612468484

Podle Tab. 47 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



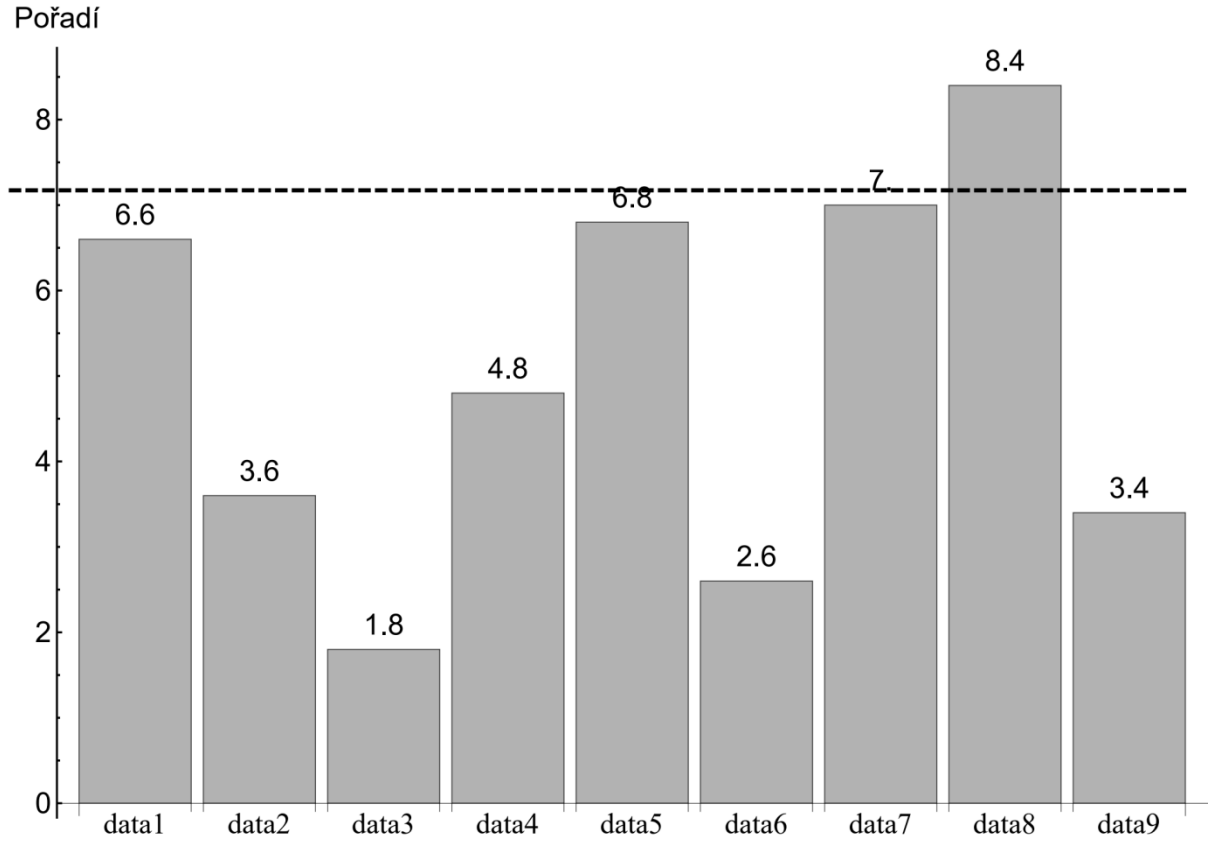
Obr. 57: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

V Obr. 57 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Tab. 48 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	9.2743362832	1.64911×10^{-6}

Podle Tab. 48 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 58: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různé kombinace technik pro úpravu dat v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

V Obr. 58 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Výsledky poukazují na významně horší výkon v detekci kybernetického útoku pro techniky spadající do oblasti data8 oproti ostatním.

OCSVM – souhrnné výsledky pro různé techniky pro úpravu datasetů

Tab. 49 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro první dva kybernetické útoky – OCSVM (dataset 1). [vlastní zdroj]

	CA1_1					CA1_2				
	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Aritmetický průměr; normalizace <0,1>	0.196	0.110	0.113	0.522	4.078	0.713	0.252	0.747	0.424	5.230
Aritmetický průměr; normalizace <-1,1>	0.213	0.190	0.120	0.671	4.136	0.721	0.270	0.753	0.413	5.361
Aritmetický průměr; standardizace	0.170	0.091	0.093	0.896	3.999	0.740	0.314	0.769	0.390	5.181
Medián; normalizace <0,1>	0.181	0.080	0.104	0.577	4.095	0.714	0.254	0.747	0.422	5.820
Medián;	0.213	0.190	0.120	0.671	3.989	0.721	0.273	0.755	0.410	5.234

normalizace <-1,1>										
Medián; standardizace	0.170	0.091	0.093	0.896	4.009	0.740	0.315	0.770	0.389	5.248
Náhrada konstantou; normalizace <0,1>	0.157	0.008	0.085	0.999	3.984	0.714	0.256	0.749	0.419	5.223
Náhrada konstantou; normalizace <-1,1>	0.013	0.000	0.007	0.085	4.091	0.721	0.271	0.754	0.412	5.351
Náhrada konstantou; standardizace	0.013	0.000	0.007	0.085	3.899	0.730	0.295	0.763	0.396	5.061

Tab. 50 – Výsledky pro jednotlivé kombinace technik pro úpravu dat pro třetí a čtvrtý kybernetický útok – OCSVM (dataset 1). [vlastní zdroj]

	CA1_3					**CA1_4**				
	M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Aritmetický průměr; normalizace <0,1>	0.079	0.107	0.042	0.406	9.423	0.002	0.005	0.001	0.412	31.358
Aritmetický průměr; normalizace <-1,1>	0.080	0.109	0.042	0.399	9.582	0.004	0.030	0.002	0.406	32.604
Aritmetický průměr; standardizace	0.089	0.137	0.047	0.400	9.314	0.004	0.037	0.002	0.400	31.152
Medián; normalizace <0,1>	0.079	0.107	0.042	0.405	9.541	0.002	0.005	0.001	0.411	32.526
Medián; normalizace <-1,1>	0.080	0.110	0.042	0.398	9.261	0.004	0.030	0.002	0.405	31.327
Medián; standardizace	0.089	0.137	0.047	0.400	9.267	0.004	0.037	0.002	0.400	31.318
Náhrada konstantou; normalizace <0,1>	0.079	0.107	0.041	0.406	9.269	0.002	0.005	0.001	0.411	31.222
Náhrada konstantou; normalizace <-1,1>	0.079	0.108	0.042	0.403	9.655	0.004	0.024	0.002	0.408	32.084
Náhrada konstantou; standardizace	0.090	0.138	0.047	0.396	9.109	0.005	0.037	0.002	0.396	30.566

Příloha B: Porovnání algoritmů strojového učení při základním nastavení.

Dataset 1 – srovnání algoritmů strojového učení při základním nastavení

Tab. 51 – Základní srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku – (dataset 1). [vlastní zdroj]

		CA1_1					CA1_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.672	0.651	0.752	0.019	0.010	0.823	0.619	0.926	0.111	0.013
	Max	0.971	0.968	0.991	0.069	0.016	0.990	0.971	0.997	0.508	0.020
	Min	0.157	0.084	0.167	0.001	0.008	0.587	0.188	0.716	0.005	0.011
LSTM	**Průměr**	0.885	0.875	0.911	0.008	0.011	0.958	0.899	0.982	0.031	0.021
	Max	0.970	0.968	0.991	0.091	0.012	0.997	0.993	0.999	0.875	0.023
	Min	0.017	-0.075	0.017	0.001	0.009	0.493	-0.338	0.548	0.002	0.016
Isolation forest	**Průměr**	0.391	0.337	0.304	0.132	0.067	0.743	0.549	0.964	0.041	0.076
	Max	0.582	0.542	0.543	0.227	0.074	0.798	0.629	0.996	0.082	0.083
	Min	0.270	0.199	0.180	0.049	0.065	0.714	0.496	0.932	0.005	0.074
OCSVM	**Výsledky**	0.213	0.190	0.120	0.671	4.136	0.721	0.270	0.753	0.413	5.361

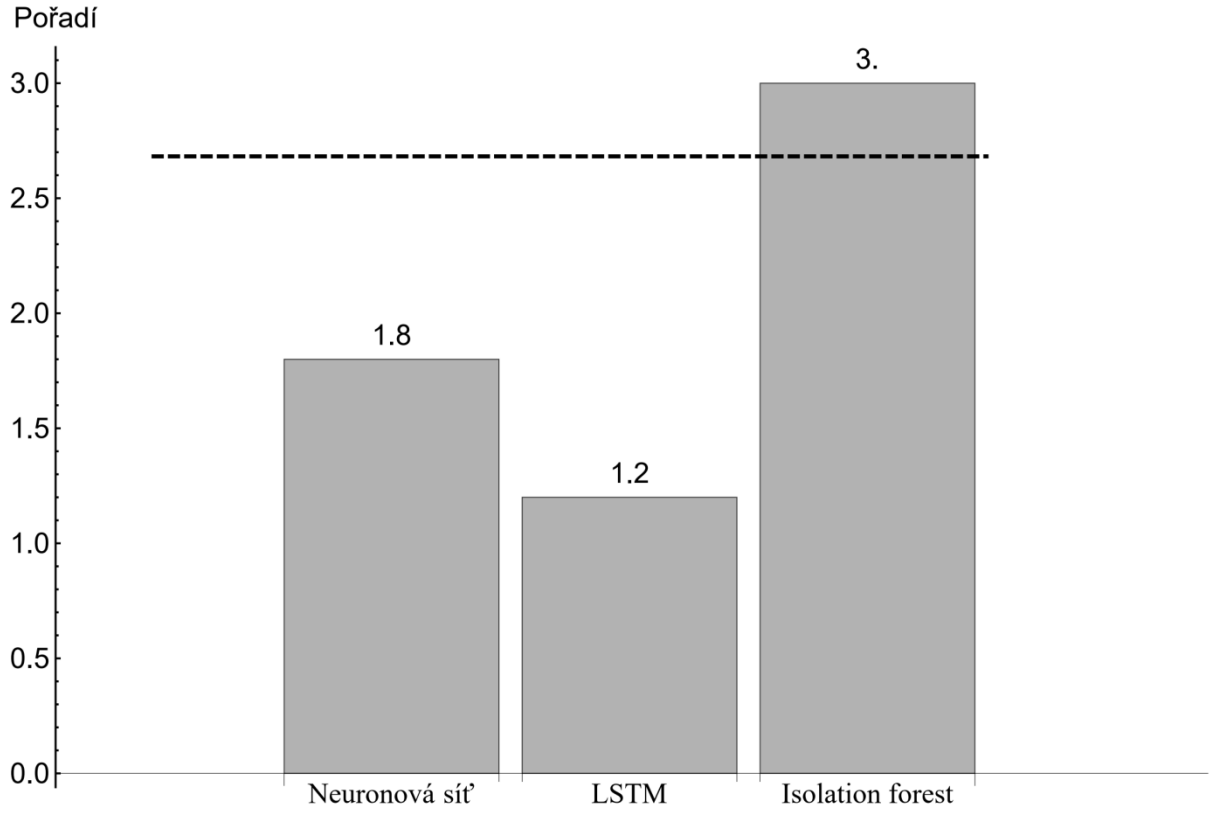
Tab. 52 – Základní srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku – (dataset 1). [vlastní zdroj]

		CA1_3					CA1_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.590	0.587	0.678	0.007	0.041	0.053	0.056	0.087	0.008	0.063
	Max	0.905	0.907	1.000	0.061	0.053	0.333	0.447	1.000	0.143	0.071
	Min	0.000	0.038	0.000	0.000	0.038	0.000	-0.012	0.000	0.000	0.059
LSTM	**Průměr**	0.722	0.724	0.825	0.003	0.350	0.276	0.280	0.336	0.004	0.069
	Max	0.966	0.965	1.000	0.036	0.417	0.824	0.837	1.000	0.248	0.093
	Min	0.000	-0.029	0.000	0.000	0.336	0.000	-0.017	0.000	0.000	0.062
Isolation forest	**Průměr**	0.145	0.125	0.118	0.043	0.118	0.002	0.000	0.001	0.045	0.425
	Max	0.343	0.330	0.369	0.096	0.131	0.027	0.094	0.014	0.094	0.460
	Min	0.022	-0.017	0.014	0.010	0.115	0.000	-0.010	0.000	0.017	0.417
OCSVM	**Výsledky**	0.080	0.109	0.042	0.399	9.582	0.004	0.030	0.002	0.406	32.604

Tab. 53 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	21.	0.00065536

Podle Tab. 53 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 % procent, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 59: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

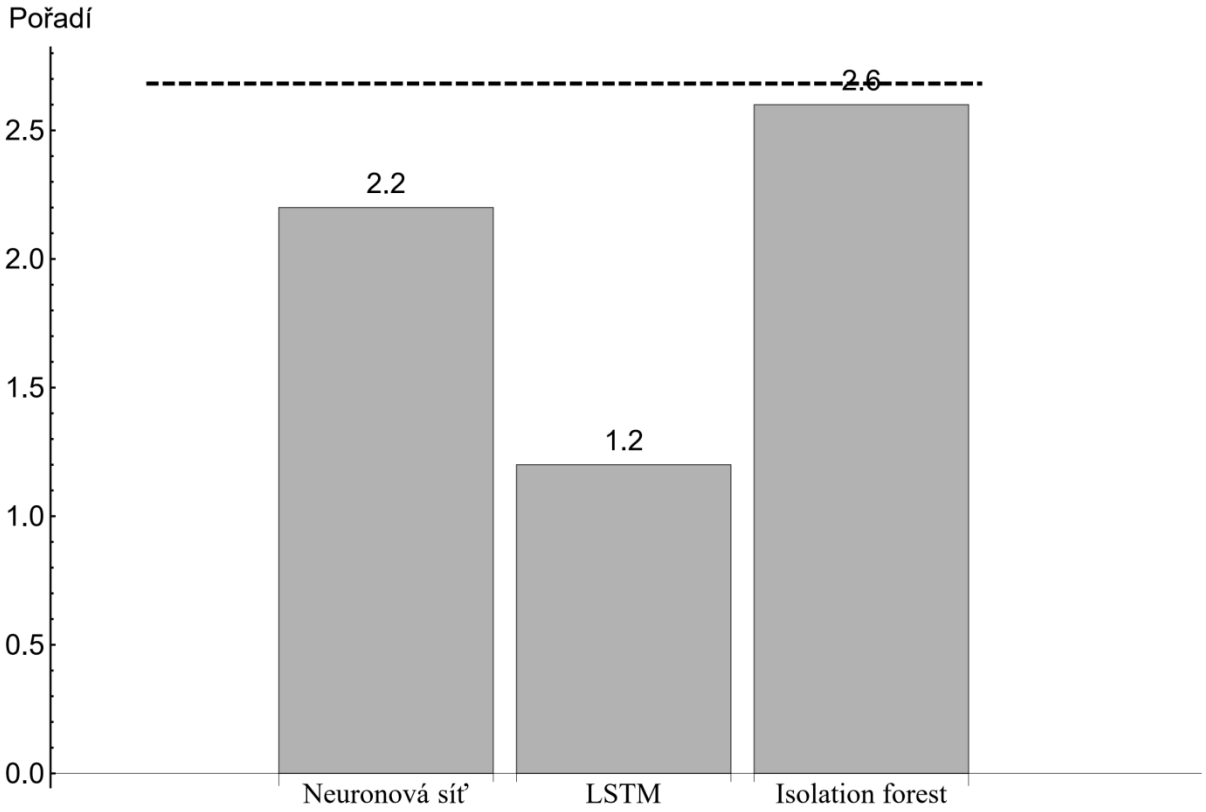
Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA1_1“ je zobrazeno v Obr. 59. Je využito Friedmanova

testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Výsledky poukazují na významně horší výkon v oblasti detekci kybernetického útoku pro algoritmus Isolation forest oproti ostatním algoritmům.

Tab. 54 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	4.33333	0.05308416

Podle Tab. 54 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



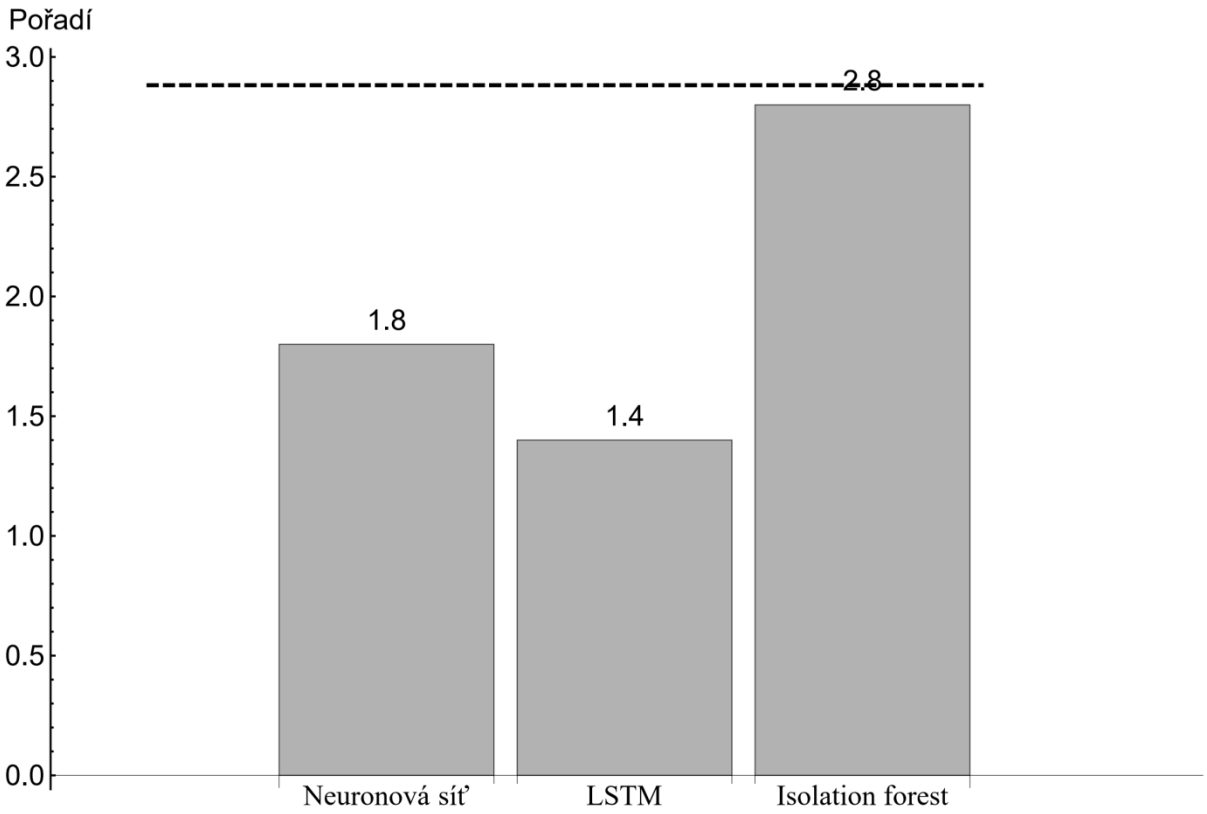
Obr. 60: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA1_2“ je zobrazeno na Obr. 60. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 55 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	4.33333	0.05308416

Podle Tab. 55 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



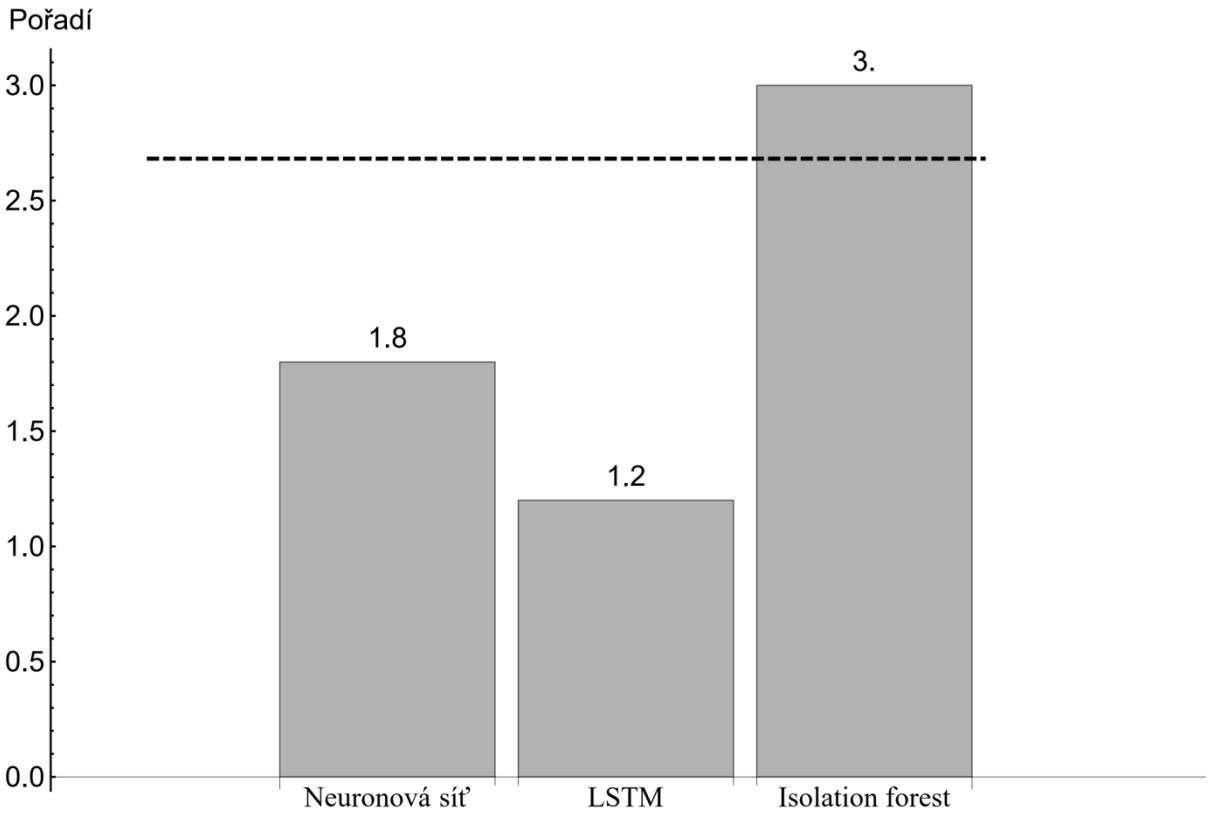
Obr. 61: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA1_2“ je zobrazeno na Obr. 61. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 56 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	21.	0.00065536

Podle Tab. 56 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 62: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA1_1“ je zobrazeno na Obr. 62. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Výsledky poukazují na významně horší výkon v oblasti detekce kybernetického útoku pro algoritmus Isolation forest oproti ostatním algoritmům.

Dataset 2 – srovnání algoritmů strojového učení při základním nastavení

Tab. 57 – Základní srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku – (dataset 2). [vlastní zdroj]

		CA2_1					CA2_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.281	0.004	0.285	0.273	2.368	0.229	0.011	0.234	0.214	2.373
	Max	0.369	0.133	0.383	0.314	2.691	0.282	0.079	0.287	0.253	2.892
	Min	0.174	-0.134	0.181	0.226	2.320	0.069	-0.190	0.071	0.181	2.323
LSTM	**Průměr**	0.271	-0.003	0.280	0.266	2.444	0.186	-0.036	0.194	0.214	2.468
	Max	0.316	0.062	0.329	0.286	2.823	0.272	0.073	0.284	0.235	3.448
	Min	0.211	-0.082	0.220	0.242	2.363	0.102	-0.132	0.111	0.187	2.373

Isolation forest	**Průměr**	0.258	0.036	0.316	0.187	15.242	0.146	-0.062	0.167	0.186	15.255
	Max	0.306	0.114	0.391	0.254	18.059	0.221	0.035	0.257	0.229	17.702
	Min	0.208	-0.065	0.230	0.153	14.510	0.079	-0.134	0.107	0.137	14.551
OCSVM	**Výsledky**	0.327	-0.071	0.252	0.546	1443.968	0.287	-0.055	0.203	0.552	1391.097

Tab. 58 – Základní srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku – (dataset 2). [vlastní zdroj]

		CA2_3					**CA2_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.246	0.048	0.251	0.195	2.375	0.268	0.087	0.273	0.178	2.370
	Max	0.349	0.176	0.351	0.218	3.166	0.336	0.171	0.341	0.201	2.710
	Min	0.157	-0.063	0.161	0.170	2.321	0.196	-0.006	0.197	0.159	2.321
LSTM	**Průměr**	0.246	0.053	0.255	0.187	2.454	0.225	0.038	0.234	0.181	2.475
	Max	0.307	0.129	0.319	0.208	2.713	0.271	0.095	0.282	0.200	3.006
	Min	0.188	-0.019	0.197	0.165	2.376	0.166	-0.033	0.175	0.165	2.379
Isolation forest	**Průměr**	0.265	0.066	0.263	0.201	15.217	0.214	0.013	0.213	0.202	15.300
	Max	0.317	0.126	0.306	0.240	17.295	0.252	0.064	0.262	0.251	21.210
	Min	0.194	-0.006	0.207	0.165	14.477	0.174	-0.035	0.175	0.147	14.512
OCSVM	**Výsledky**	0.390	0.156	0.267	0.534	1396.110	0.280	-0.028	0.192	0.550	1393.412

Tab. 59 – Základní srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku – (dataset 2). [vlastní zdroj]

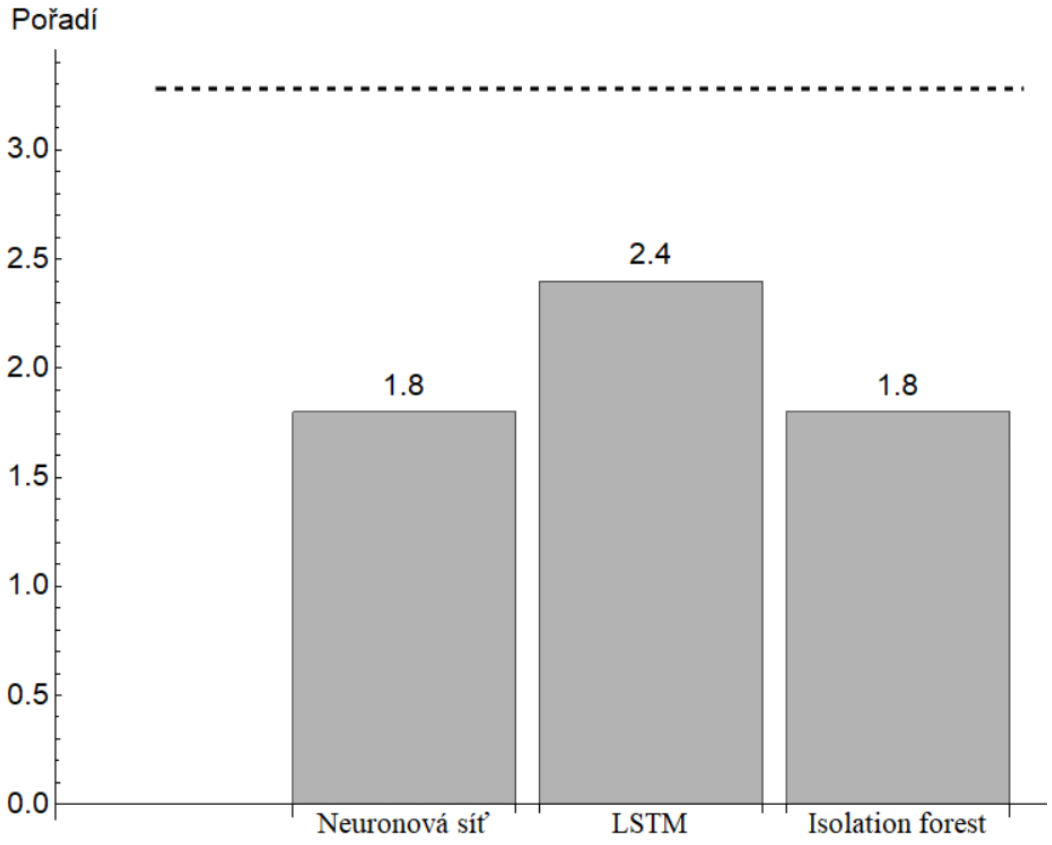
		CA2_5					**CA2_6**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.185	-0.024	0.188	0.206	2.384	0.346	0.155	0.353	0.187	2.367
	Max	0.319	0.145	0.325	0.224	2.874	0.486	0.338	0.498	0.244	2.808
	Min	0.104	-0.125	0.106	0.170	2.322	0.176	-0.069	0.177	0.143	2.315
LSTM	**Průměr**	0.203	0.006	0.212	0.189	2.461	0.238	0.023	0.249	0.206	2.456
	Max	0.300	0.130	0.318	0.213	2.814	0.289	0.092	0.308	0.231	2.993
	Min	0.106	-0.110	0.113	0.159	2.380	0.152	-0.086	0.160	0.174	2.350
Isolation forest	**Průměr**	0.272	0.108	0.306	0.146	15.265	0.204	-0.004	0.227	0.191	15.182
	Max	0.372	0.228	0.410	0.184	16.674	0.263	0.075	0.296	0.239	17.788
	Min	0.198	0.025	0.231	0.113	14.570	0.127	-0.119	0.133	0.138	14.522

OCSVM	**Výsledky**	0.283	-0.035	0.195	0.560	1389.863	0.253	-0.125	0.181	0.567	1404.976

Tab. 60 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_1. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	0.54545454	0.59969536

Podle Tab. 60 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 % procent, a tudíž mezi daty neexistuje statisticky významný rozdíl.



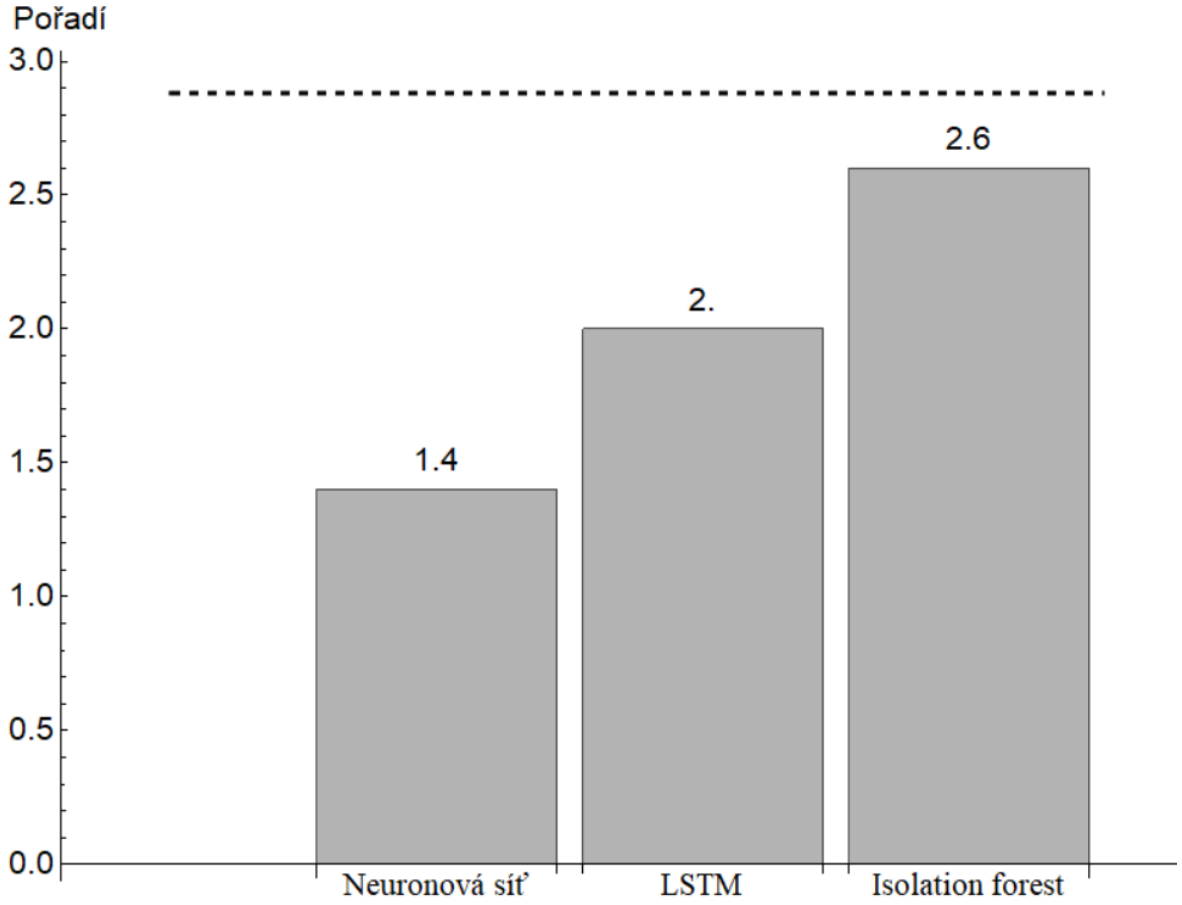
Obr. 63: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA2_1. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA2_1“ je zobrazeno na Obr. 63. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 61 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_2. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.25	0.16777216

Podle Tab. 61 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



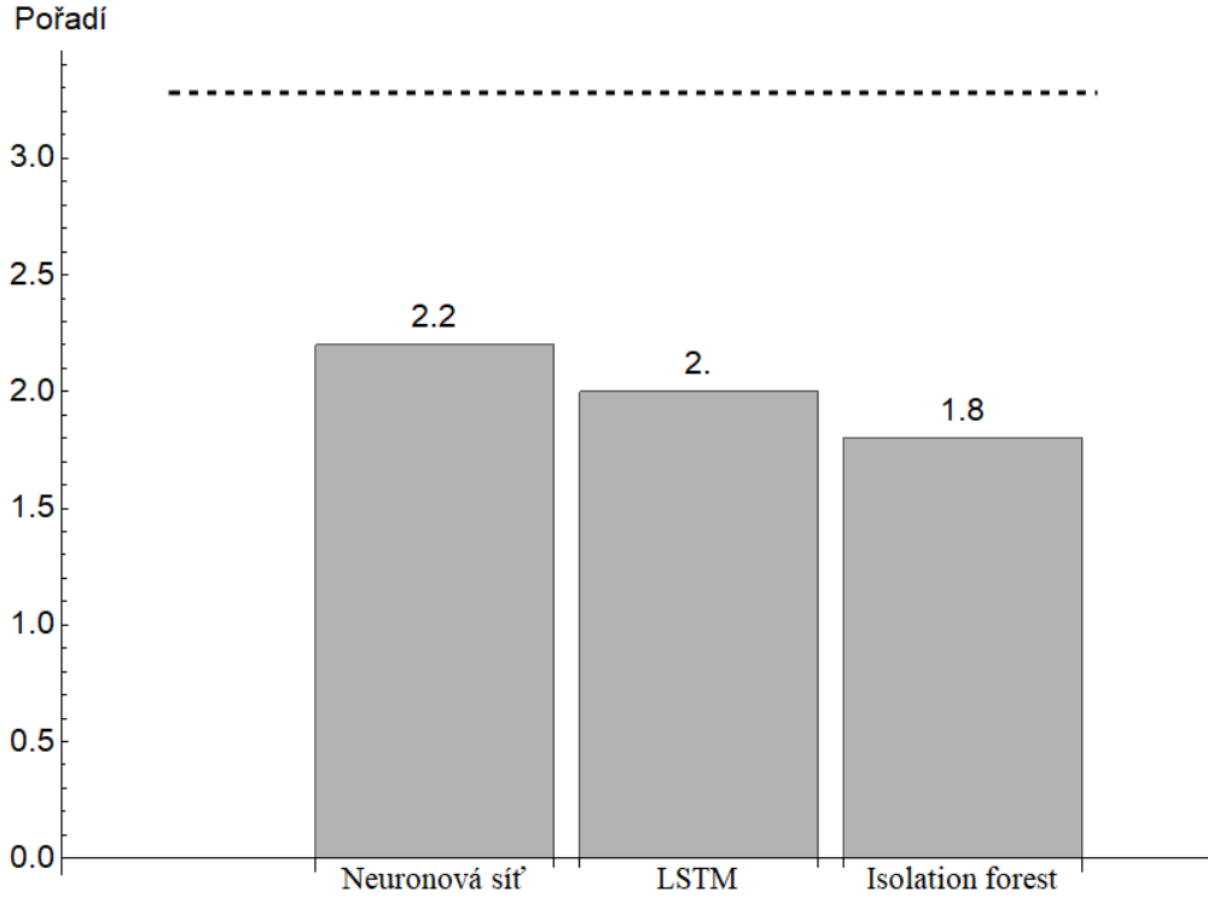
Obr. 64: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA2_2. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA2_2“ je zobrazeno na Obr. 64. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 62 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_3. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	0.16666666	0.84934656

Podle Tab. 62 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 % procent, a tudíž mezi daty neexistuje statisticky významný rozdíl.



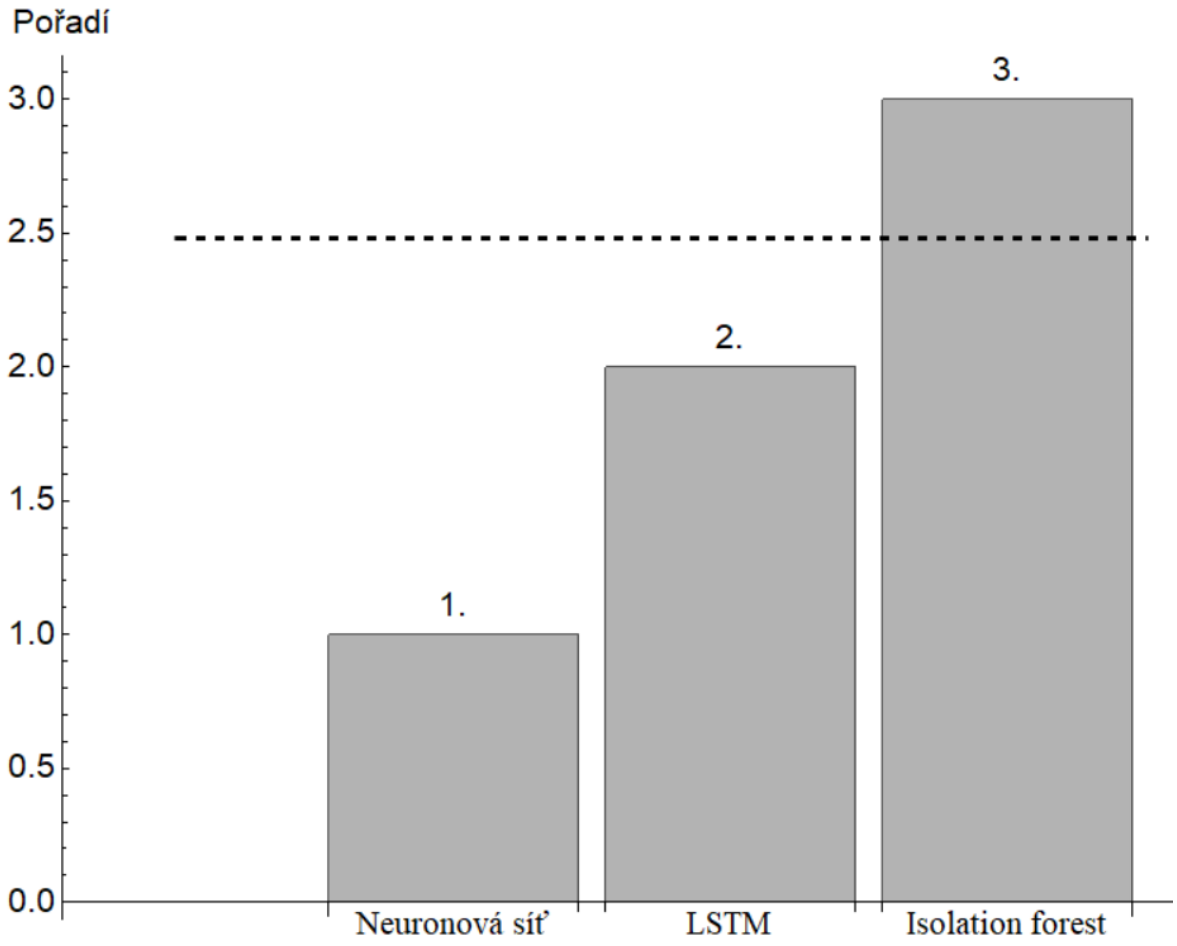
Obr. 65: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA2_3. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA2_3“ je zobrazeno na Obr. 65. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 63 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_4. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	21.	0.00054237

Podle Tab. 63 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



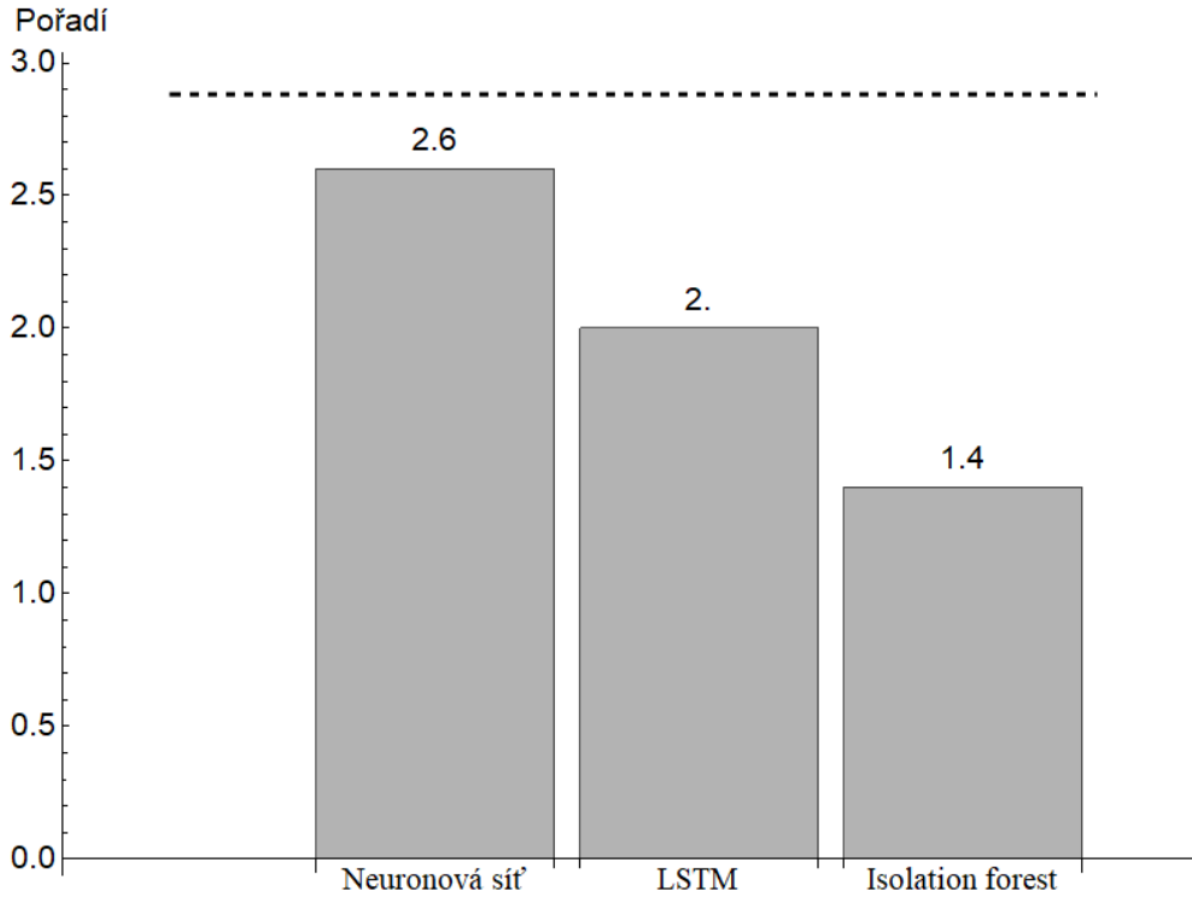
Obr. 66: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA2_4. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA2_4“ je zobrazeno na Obr. 66. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Výsledky poukazují na významně horší výkon v oblasti detekce kybernetického útoku pro algoritmus Isolation forest oproti ostatním algoritmům.

Tab. 64 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_5. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.25	0.16777216

Podle Tab. 64 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



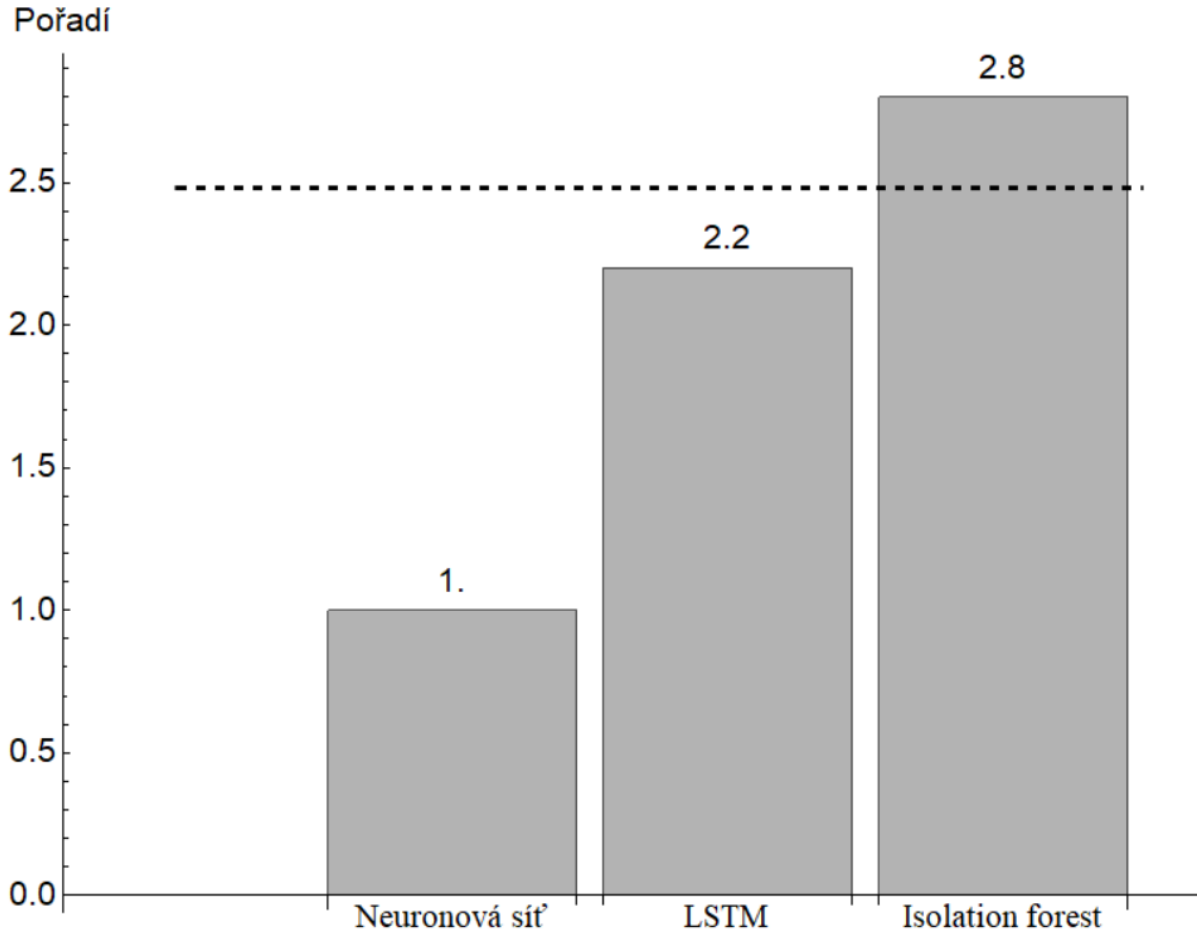
Obr. 67: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA2_5. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA2_5“ je zobrazeno na Obr. 67. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 65 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_6. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	21.	0.00065536

Podle Tab. 65 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 68: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA2_6. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA2_6“ je zobrazeno na Obr. 68. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Výsledky poukazují na významně horší výkon v oblasti detekce kybernetického útoku pro algoritmus Isolation forest oproti ostatním algoritmům.

Dataset 3 – srovnání algoritmů strojového učení při základním nastavení

Tab. 66 – Základní srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku – (dataset 3). [vlastní zdroj]

		CA3_1					CA3_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.450	0.280	0.460	0.165	0.032	0.467	0.308	0.478	0.154	0.032
	Max	0.466	0.297	0.472	0.181	0.043	0.511	0.372	0.541	0.170	0.048
	Min	0.415	0.233	0.422	0.154	0.029	0.445	0.275	0.447	0.127	0.029
LSTM	**Průměr**	0.437	0.267	0.441	0.168	0.038	0.459	0.306	0.466	0.150	0.038
	Max	0.498	0.343	0.501	0.206	0.045	0.496	0.351	0.499	0.195	0.048
	Min	0.000	-0.007	0.000	0.000	0.033	0.319	0.123	0.320	0.134	0.032
Isolation forest	**Průměr**	0.432	0.293	0.515	0.113	0.126	0.474	0.342	0.544	0.111	0.122
	Max	0.477	0.372	0.619	0.151	0.154	0.517	0.410	0.650	0.149	0.147
	Min	0.388	0.224	0.433	0.076	0.120	0.442	0.282	0.467	0.069	0.117
OCSVM	**Výsledky**	0.418	0.194	0.359	0.285	10.529	0.413	0.193	0.352	0.285	10.535

Tab. 67 – Základní srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku – (dataset 3). [vlastní zdroj]

		CA3_3					CA3_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.310	0.130	0.304	0.183	0.041	0.438	0.252	0.447	0.180	0.034
	Max	0.561	0.285	0.511	0.688	0.071	0.466	0.292	0.481	0.201	0.045
	Min	0.000	-0.029	0.000	0.001	0.035	0.396	0.195	0.402	0.165	0.030
LSTM	**Průměr**	0.373	0.201	0.375	0.164	0.046	0.423	0.239	0.425	0.182	0.038
	Max	0.519	0.307	0.521	0.246	0.061	0.644	0.327	0.563	0.750	0.047
	Min	0.000	-0.105	0.000	0.001	0.042	0.000	-0.041	0.000	0.000	0.031
Isolation forest	**Průměr**	0.546	0.398	0.656	0.112	0.134	0.243	0.092	0.357	0.114	0.125
	Max	0.578	0.454	0.737	0.143	0.230	0.311	0.186	0.466	0.147	0.147
	Min	0.472	0.323	0.592	0.075	0.126	0.202	0.031	0.287	0.083	0.120
OCSVM	**Výsledky**	0.469	0.208	0.442	0.285	11.876	0.428	0.197	0.374	0.285	11.127

Tab. 68 – Základní srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku – (dataset 3). [vlastní zdroj]

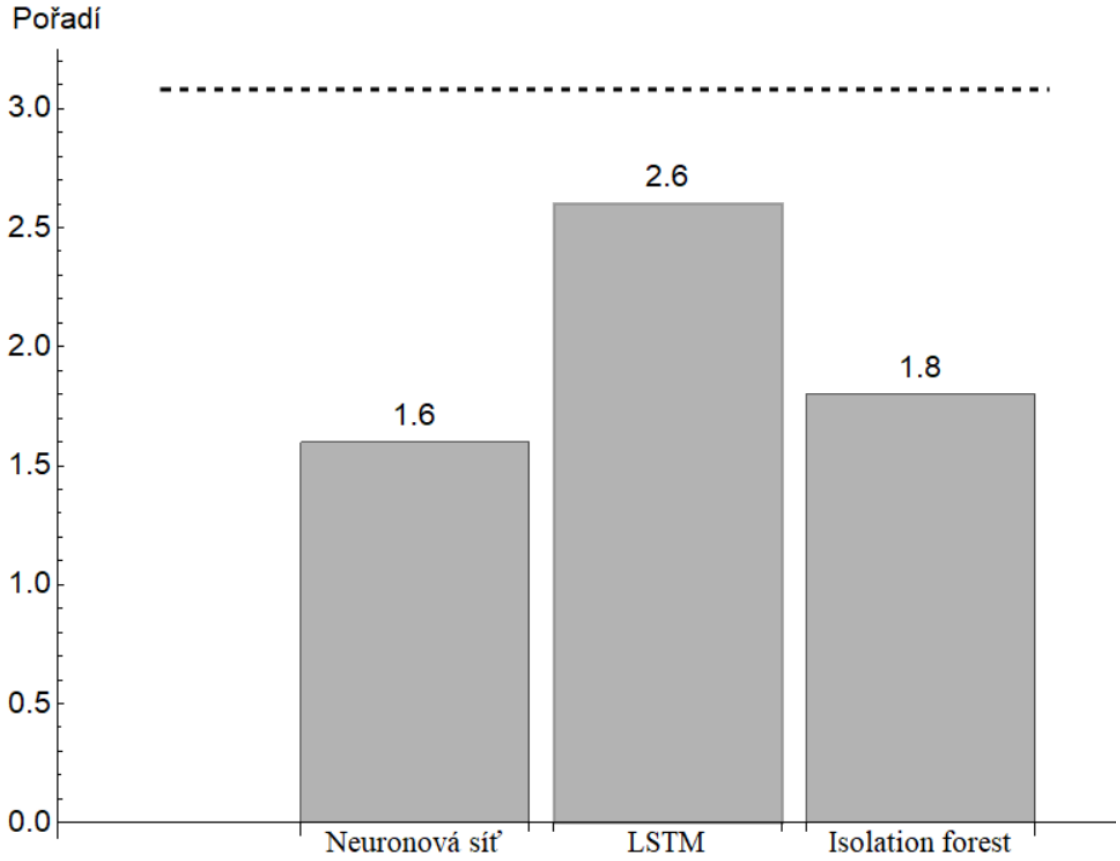
		CA3_5					CA3_6				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.509	0.361	0.518	0.144	0.036	0.503	0.270	0.507	0.231	0.045

	Max	0.527	0.392	0.557	0.152	0.079	0.513	0.290	0.526	0.231	0.107
	Min	0.502	0.350	0.505	0.123	0.029	0.502	0.268	0.506	0.214	0.033
LSTM	**Průměr**	0.428	0.295	0.437	0.125	0.042	0.516	0.296	0.522	0.216	0.049
	Max	0.513	0.377	0.526	0.203	0.087	0.574	0.378	0.576	0.232	0.087
	Min	0.000	-0.126	0.000	0.000	0.032	0.487	0.255	0.495	0.182	0.039
Isolation forest	**Průměr**	0.378	0.237	0.465	0.113	0.122	0.519	0.365	0.649	0.112	0.136
	Max	0.486	0.373	0.594	0.148	0.143	0.559	0.432	0.721	0.150	0.177
	Min	0.221	0.092	0.333	0.071	0.118	0.447	0.260	0.552	0.081	0.129
OCSVM	**Výsledky**	0.413	0.193	0.351	0.285	10.756	0.476	0.210	0.454	0.285	11.650

Tab. 69 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_1. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.55555556	0.26873856

Podle Tab. 69 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 % procent, a tudíž mezi daty neexistuje statisticky významný rozdíl.



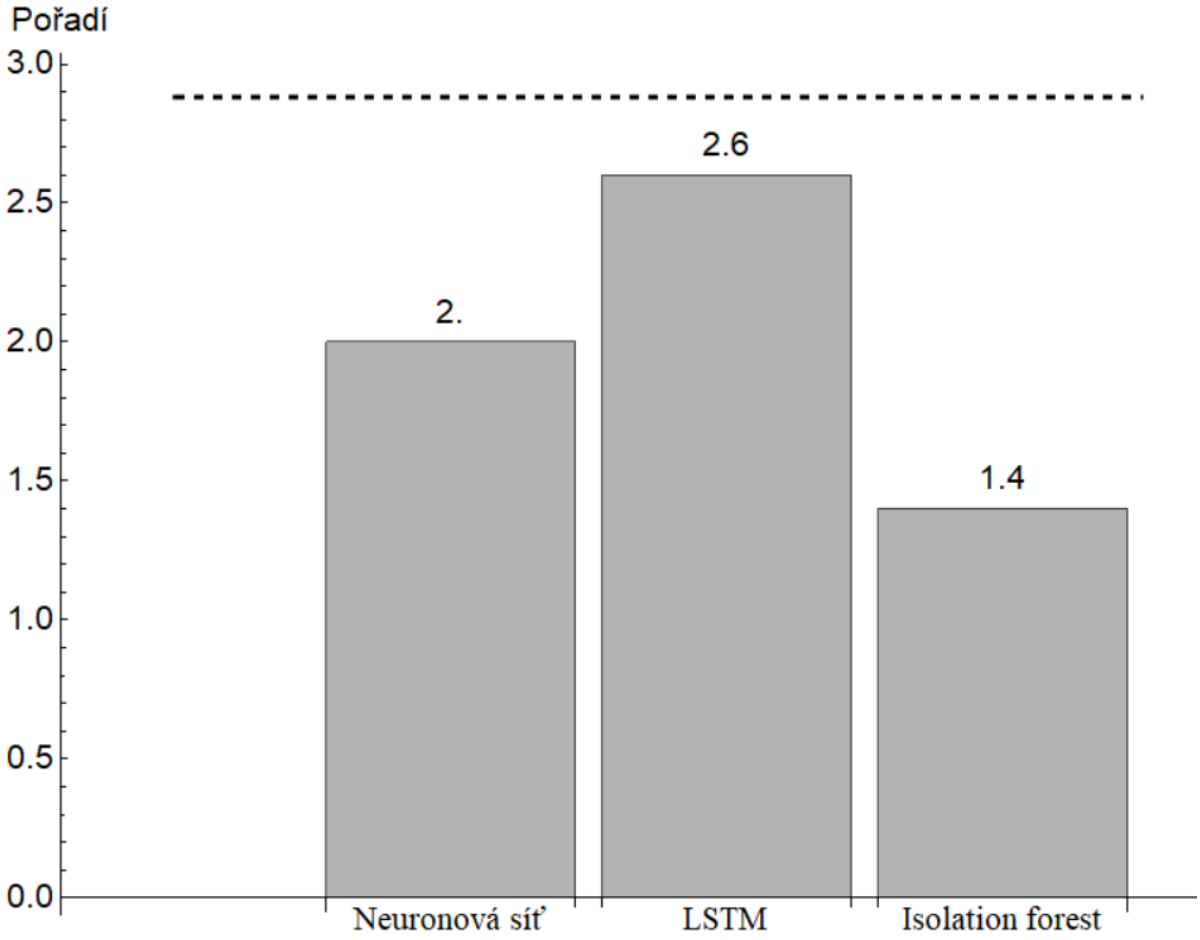
Obr. 69: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA3_1. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA3_1“ je zobrazeno na Obr. 69. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 70 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_2. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.25	0.16777216

Podle Tab. 70 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



Obr. 70: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA3_2. [vlastní zdroj]

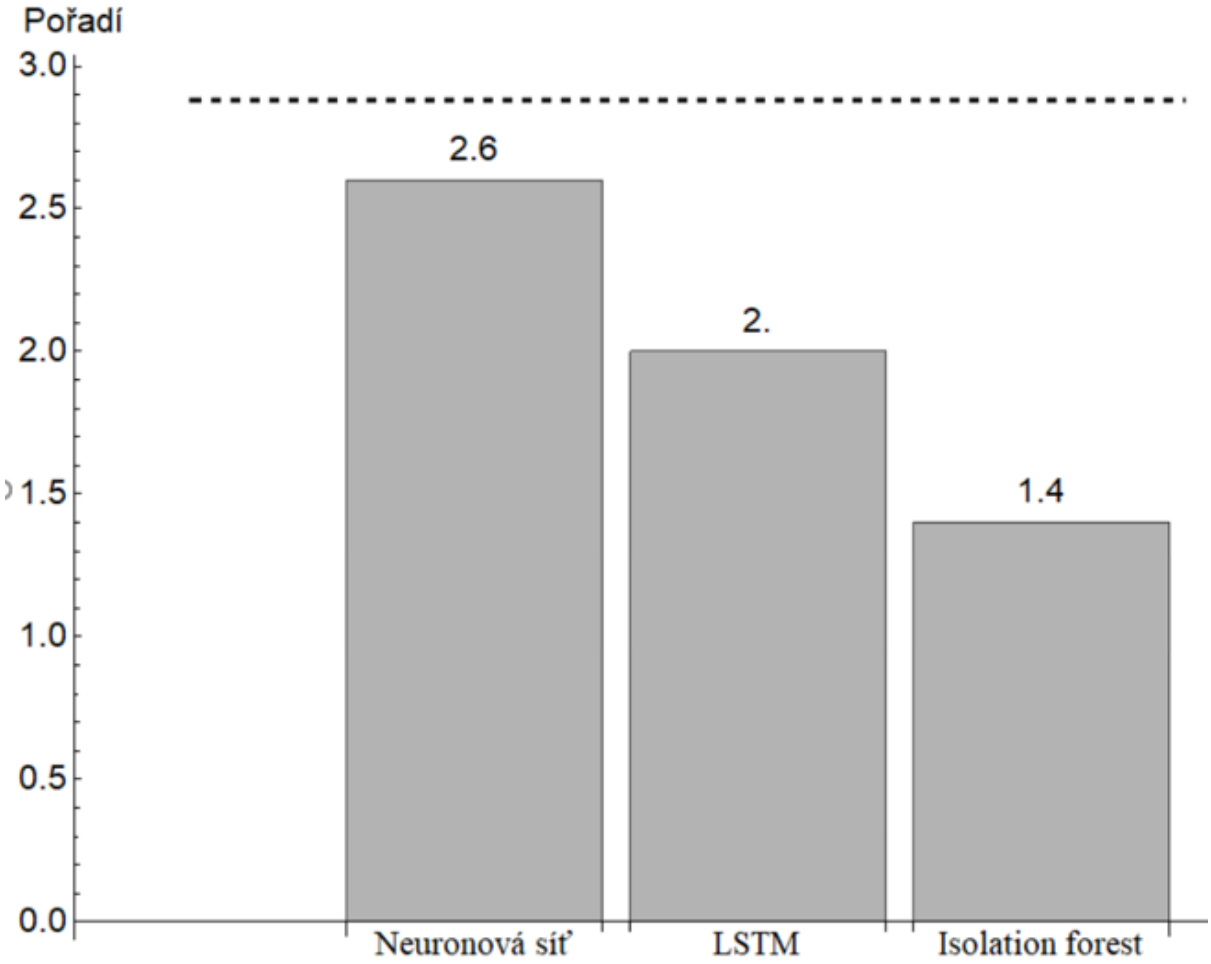
Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA3_2“ je zobrazeno na Obr. 70. Je využito Friedmanova testu a Nemenyiho

kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 71 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_3. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.25	0.16777216

Podle Tab. 71 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



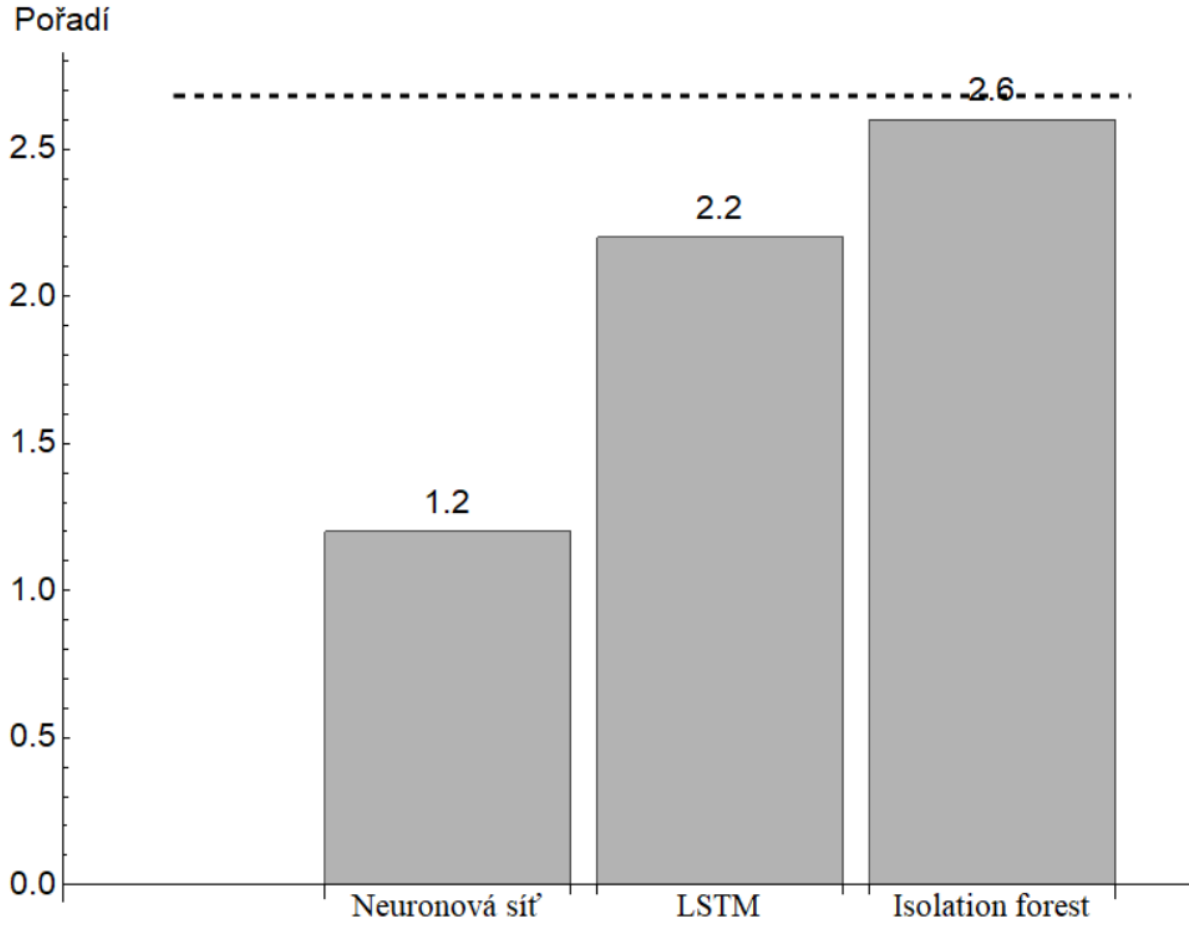
Obr. 71: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA3_3. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA3_3“ je zobrazeno na Obr. 71. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 72 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_4. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	4.33333333	0.05308416

Podle Tab. 72 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



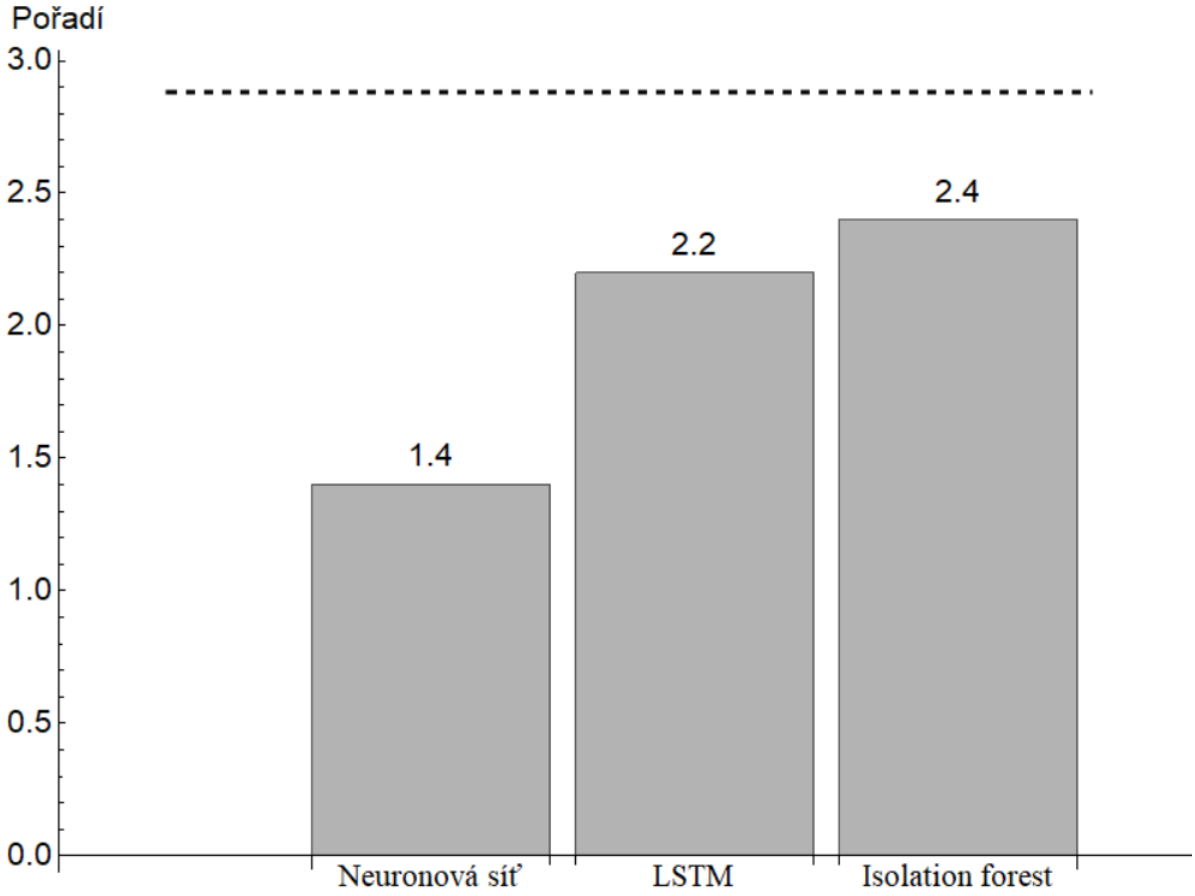
Obr. 72: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA3_4. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA3_4“ je zobrazeno na Obr. 72. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 73 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_5. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.55555556	0.26873856

Podle Tab. 73 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



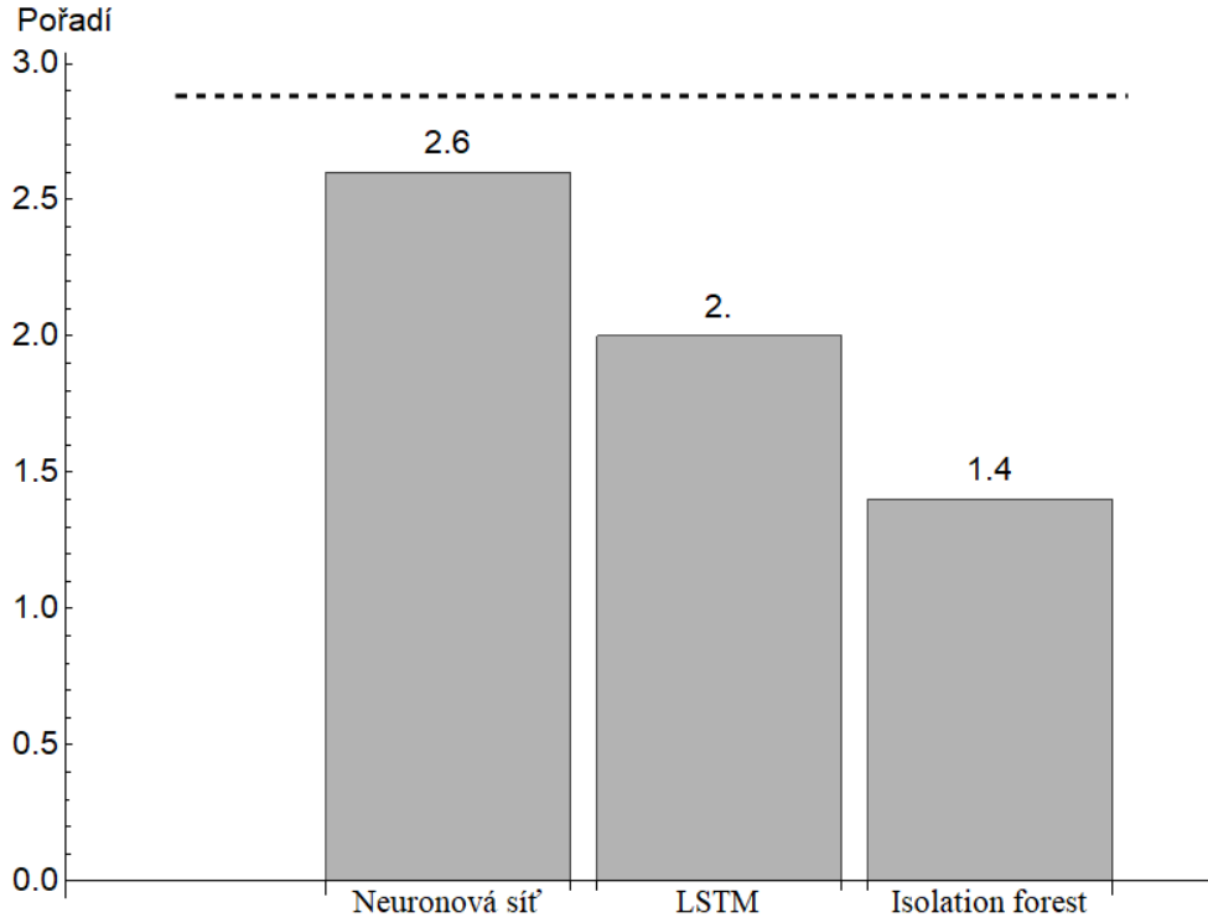
Obr. 73: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA3_5. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA3_5“ je zobrazeno na Obr. 73. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Tab. 74 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_6. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.25	0.16777216

Podle Tab. 74 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



Obr. 74: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro vybrané stochastické algoritmy v rámci kybernetického útoku – CA3_6. [vlastní zdroj]

Srovnání vybraných algoritmů strojového učení v rámci kybernetického útoku „CA3_6“ je zobrazeno na Obr. 74. Je využito Friedmanova testu a Nemenyiho kritické vzdálenosti k porovnání algoritmů. Podle výsledků nelze konstatovat, který z algoritmů je nejlepší.

Příloha C: Výsledky algoritmu OCSVM pro rozdílné hodnoty gamma parametru v rámci tří datasetů.

Tab. 75 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro první a druhý kybernetický útok – (dataset 1). [vlastní zdroj]

	CA1_1					CA1_2				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.144	0.002	0.086	0.442	4.386	0.633	0.129	0.705	0.439	5.796
0.1	0.113	-0.104	0.064	0.665	3.986	0.638	0.152	0.716	0.416	5.214
0.15	0.185	0.089	0.106	0.561	3.987	0.668	0.191	0.730	0.416	5.210
0.2	0.213	0.190	0.120	0.671	3.981	0.721	0.270	0.753	0.413	5.253
0.25	0.206	0.175	0.115	0.704	4.044	0.722	0.277	0.757	0.405	5.269
0.3	0.171	0.094	0.094	0.890	4.391	0.720	0.267	0.752	0.416	5.672
0.35	0.171	0.094	0.094	0.890	4.267	0.718	0.260	0.749	0.424	5.443
0.4	0.171	0.093	0.094	0.892	4.609	0.725	0.270	0.751	0.424	5.482
0.45	0.171	0.092	0.093	0.893	3.992	0.887	0.648	0.806	0.433	5.234
0.5	0.171	0.092	0.093	0.893	4.354	0.885	0.641	0.802	0.444	5.711
0.55	0.171	0.092	0.093	0.893	3.979	0.891	0.667	0.804	0.447	5.235
0.6	0.171	0.092	0.093	0.893	3.994	0.889	0.661	0.801	0.454	5.227
0.65	0.171	0.092	0.093	0.893	3.983	0.893	0.673	0.806	0.439	5.209
0.7	0.171	0.092	0.093	0.893	3.992	0.892	0.669	0.805	0.444	5.228
0.75	0.171	0.092	0.093	0.893	3.977	0.890	0.662	0.801	0.453	5.215
0.8	0.171	0.092	0.093	0.893	3.998	0.890	0.663	0.802	0.451	5.206
0.85	0.171	0.092	0.093	0.893	3.974	0.890	0.663	0.802	0.451	5.217
0.9	0.171	0.092	0.093	0.893	4.418	0.889	0.660	0.800	0.456	5.823
0.95	0.171	0.092	0.093	0.893	3.989	0.889	0.661	0.801	0.454	5.232
1	0.171	0.092	0.093	0.893	3.988	0.888	0.655	0.798	0.462	5.245

Tab. 76 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro třetí a čtvrtý kybernetický útok – (dataset 1). [vlastní zdroj]

	CA1_3					CA1_4				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.006	-0.111	0.003	0.421	10.894	0.001	-0.007	0.001	0.416	34.225
0.1	0.006	-0.106	0.003	0.403	9.241	0.002	0.000	0.001	0.404	31.122
0.15	0.018	-0.072	0.009	0.396	9.241	0.002	0.006	0.001	0.399	31.163
0.2	0.080	0.109	0.042	0.399	9.247	0.004	0.030	0.002	0.406	31.200
0.25	0.080	0.109	0.042	0.400	9.314	0.004	0.036	0.002	0.412	32.163
0.3	0.080	0.109	0.042	0.401	9.812	0.004	0.035	0.002	0.416	32.622
0.35	0.078	0.105	0.041	0.410	9.704	0.004	0.035	0.002	0.421	32.476
0.4	0.087	0.135	0.046	0.416	9.708	0.004	0.035	0.002	0.423	31.270
0.45	0.086	0.134	0.045	0.422	9.275	0.004	0.034	0.002	0.431	31.190
0.5	0.085	0.131	0.045	0.428	10.434	0.004	0.034	0.002	0.437	34.008

0.55	0.093	0.158	0.049	0.433	9.249	0.004	0.034	0.002	0.441	31.067
0.6	0.092	0.157	0.048	0.436	9.213	0.004	0.033	0.002	0.444	31.128
0.65	0.091	0.155	0.048	0.442	9.227	0.004	0.033	0.002	0.447	31.067
0.7	0.090	0.153	0.047	0.447	9.233	0.004	0.033	0.002	0.448	31.114
0.75	0.089	0.152	0.047	0.452	9.257	0.004	0.033	0.002	0.451	31.084
0.8	0.089	0.151	0.047	0.454	9.222	0.004	0.033	0.002	0.453	31.091
0.85	0.088	0.150	0.046	0.458	9.238	0.004	0.033	0.002	0.456	31.072
0.9	0.088	0.149	0.046	0.461	10.262	0.004	0.033	0.002	0.458	34.461
0.95	0.089	0.155	0.047	0.465	9.247	0.004	0.032	0.002	0.463	31.139
1	0.090	0.158	0.047	0.468	9.255	0.004	0.032	0.002	0.465	31.137

Tab. 77 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro první a druhý kybernetický útok – (dataset 2). [vlastní zdroj]

	CA2_1					**CA2_2**				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.332	-0.055	0.258	0.528	1842.133	0.292	-0.040	0.209	0.533	1912.737
0.1	0.330	-0.062	0.255	0.536	1389.618	0.290	-0.045	0.206	0.540	1388.851
0.15	0.328	-0.068	0.253	0.543	1389.579	0.288	-0.051	0.205	0.548	1388.454
0.2	0.327	-0.071	0.252	0.546	1388.888	0.287	-0.055	0.203	0.552	1387.812
0.25	0.326	-0.073	0.251	0.547	1389.120	0.285	-0.058	0.202	0.554	1389.590
0.3	0.326	-0.074	0.251	0.549	1389.614	0.284	-0.061	0.201	0.558	1390.277
0.35	0.326	-0.074	0.251	0.549	1494.931	0.284	-0.061	0.201	0.559	1730.380
0.4	0.327	-0.072	0.251	0.549	1396.182	0.286	-0.059	0.202	0.559	1388.961
0.45	0.328	-0.072	0.252	0.549	1389.664	0.287	-0.057	0.202	0.559	1388.446
0.5	0.328	-0.072	0.252	0.549	1390.430	0.288	-0.056	0.203	0.559	1544.310
0.55	0.328	-0.071	0.252	0.550	1409.449	0.288	-0.054	0.204	0.558	1415.694
0.6	0.329	-0.070	0.252	0.549	1780.423	0.289	-0.054	0.204	0.558	1826.832
0.65	0.330	-0.068	0.253	0.549	1419.547	0.289	-0.052	0.204	0.558	1467.599
0.7	0.331	-0.066	0.254	0.549	1392.744	0.290	-0.052	0.204	0.559	1390.569
0.75	0.332	-0.064	0.255	0.549	1391.798	0.290	-0.053	0.204	0.561	1390.169
0.8	0.334	-0.062	0.256	0.550	1447.686	0.290	-0.053	0.204	0.563	1495.271
0.85	0.334	-0.063	0.256	0.552	1390.610	0.292	-0.051	0.205	0.565	1390.126
0.9	0.336	-0.062	0.256	0.556	1415.775	0.293	-0.051	0.205	0.569	1471.392
0.95	0.338	-0.061	0.257	0.561	1605.881	0.295	-0.047	0.207	0.572	1628.993
1	0.339	-0.062	0.257	0.566	1769.527	0.299	-0.041	0.209	0.574	1836.380

Tab. 78 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro třetí a čtvrtý kybernetický útok – (dataset 2). [vlastní zdroj]

	CA2_3					CA2_4				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.375	0.128	0.257	0.536	1868.875	0.284	-0.018	0.196	0.539	1756.667
0.1	0.381	0.140	0.261	0.535	1389.299	0.282	-0.022	0.194	0.544	1389.481
0.15	0.387	0.151	0.265	0.535	1388.335	0.281	-0.026	0.193	0.549	1388.396
0.2	0.390	0.156	0.267	0.534	1387.767	0.280	-0.028	0.192	0.550	1387.870
0.25	0.392	0.160	0.268	0.534	1389.904	0.279	-0.029	0.192	0.551	1389.109
0.3	0.392	0.161	0.269	0.533	1388.874	0.278	-0.031	0.191	0.553	1389.160
0.35	0.393	0.162	0.269	0.532	1774.171	0.278	-0.033	0.190	0.553	1650.683
0.4	0.392	0.161	0.269	0.532	1389.182	0.277	-0.034	0.190	0.553	1388.161
0.45	0.393	0.162	0.269	0.532	1388.778	0.277	-0.035	0.190	0.553	1388.665
0.5	0.393	0.162	0.269	0.532	1516.823	0.276	-0.036	0.189	0.554	1611.793
0.55	0.392	0.161	0.269	0.532	1455.231	0.275	-0.037	0.189	0.555	1640.361
0.6	0.392	0.160	0.268	0.532	1839.733	0.275	-0.038	0.188	0.555	1734.012
0.65	0.392	0.160	0.268	0.532	1645.921	0.274	-0.039	0.188	0.555	1754.972
0.7	0.391	0.159	0.268	0.533	1390.418	0.274	-0.041	0.188	0.556	1390.089
0.75	0.391	0.158	0.267	0.534	1391.176	0.273	-0.042	0.187	0.557	1390.801
0.8	0.390	0.157	0.267	0.536	1408.071	0.272	-0.045	0.186	0.559	1390.807
0.85	0.392	0.160	0.268	0.537	1390.600	0.270	-0.049	0.185	0.563	1390.300
0.9	0.391	0.158	0.267	0.540	1513.197	0.268	-0.054	0.183	0.569	1431.238
0.95	0.390	0.156	0.265	0.546	1695.407	0.266	-0.059	0.181	0.576	1891.251
1	0.389	0.153	0.264	0.551	1777.975	0.267	-0.060	0.181	0.581	1414.874

Tab. 79 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro pátý a šestý kybernetický útok – (dataset 2). [vlastní zdroj]

	CA2_5					CA2_6				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.294	-0.015	0.202	0.559	1870.658	0.259	-0.115	0.185	0.567	1510.359
0.1	0.290	-0.022	0.200	0.559	1389.013	0.256	-0.119	0.183	0.568	1388.202
0.15	0.286	-0.029	0.197	0.560	1388.965	0.255	-0.123	0.182	0.569	1388.772
0.2	0.283	-0.035	0.195	0.560	1388.903	0.253	-0.125	0.181	0.567	1388.566
0.25	0.279	-0.043	0.192	0.561	1389.895	0.251	-0.126	0.180	0.566	1389.506
0.3	0.276	-0.047	0.190	0.562	1389.156	0.250	-0.128	0.179	0.566	1388.837
0.35	0.274	-0.050	0.189	0.561	1690.438	0.250	-0.129	0.179	0.566	1739.426
0.4	0.273	-0.052	0.189	0.561	1388.635	0.250	-0.129	0.179	0.566	1390.233
0.45	0.272	-0.054	0.188	0.561	1389.040	0.249	-0.129	0.179	0.565	1389.589
0.5	0.272	-0.054	0.188	0.561	1765.675	0.249	-0.130	0.178	0.565	1651.808
0.55	0.272	-0.054	0.188	0.560	1561.797	0.248	-0.131	0.178	0.565	1569.969
0.6	0.271	-0.055	0.187	0.559	1740.931	0.247	-0.133	0.177	0.566	1463.998
0.65	0.271	-0.054	0.188	0.558	1778.505	0.246	-0.136	0.176	0.567	1770.314

0.7	0.271	-0.054	0.187	0.557	1390.488	0.244	-0.139	0.175	0.569	1389.991
0.75	0.271	-0.055	0.187	0.557	1389.733	0.242	-0.143	0.173	0.572	1390.762
0.8	0.270	-0.056	0.187	0.557	1390.941	0.240	-0.149	0.172	0.576	1390.751
0.85	0.270	-0.057	0.186	0.557	1390.833	0.238	-0.154	0.170	0.579	1391.301
0.9	0.269	-0.058	0.186	0.559	1407.006	0.235	-0.161	0.168	0.585	1406.853
0.95	0.268	-0.062	0.185	0.562	1722.675	0.233	-0.167	0.166	0.591	1998.158
1	0.265	-0.068	0.183	0.568	1409.695	0.231	-0.173	0.164	0.596	1409.432

Tab. 80 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro první a druhý kybernetický útok – (dataset 3). [vlastní zdroj]

	CA3_1					**CA3_2**				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.429	0.216	0.376	0.264	10.597	0.425	0.215	0.369	0.264	10.596
0.1	0.420	0.200	0.363	0.280	10.641	0.416	0.198	0.356	0.280	10.620
0.15	0.288	-0.113	0.202	0.630	10.685	0.283	-0.112	0.197	0.630	10.810
0.2	0.418	0.194	0.359	0.285	10.423	0.413	0.193	0.352	0.285	10.388
0.25	0.285	-0.124	0.199	0.641	10.684	0.280	-0.123	0.194	0.641	11.100
0.3	0.421	0.181	0.267	0.878	11.197	0.413	0.179	0.261	0.878	10.927
0.35	0.545	0.419	0.375	0.532	10.496	0.537	0.415	0.368	0.532	10.439
0.4	0.421	0.180	0.266	0.879	10.369	0.413	0.178	0.260	0.879	10.377
0.45	0.399	0.160	0.332	0.320	10.360	0.394	0.159	0.326	0.320	10.351
0.5	0.544	0.417	0.374	0.535	10.397	0.536	0.413	0.367	0.535	10.330
0.55	0.279	-0.146	0.193	0.665	10.358	0.274	-0.145	0.189	0.665	10.330
0.6	0.544	0.417	0.373	0.535	10.352	0.536	0.413	0.366	0.535	10.362
0.65	0.420	0.178	0.266	0.881	10.569	0.413	0.176	0.260	0.881	10.437
0.7	0.539	0.409	0.369	0.546	10.345	0.531	0.405	0.362	0.546	10.318
0.75	0.539	0.409	0.369	0.547	10.440	0.531	0.405	0.361	0.547	10.416
0.8	0.276	-0.158	0.190	0.678	10.236	0.271	-0.157	0.186	0.678	10.160
0.85	0.277	-0.155	0.191	0.675	10.200	0.272	-0.154	0.187	0.675	10.156
0.9	0.548	0.422	0.377	0.527	10.189	0.540	0.418	0.370	0.527	10.162
0.95	0.281	-0.140	0.195	0.658	10.186	0.276	-0.138	0.190	0.658	10.448
1	0.281	-0.139	0.195	0.658	10.199	0.276	-0.138	0.190	0.658	10.324

Tab. 81 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro třetí a čtvrtý kybernetický útok – (dataset 3). [vlastní zdroj]

	CA3_3					**CA3_4**				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.480	0.231	0.461	0.264	12.177	0.439	0.219	0.392	0.264	10.716
0.1	0.472	0.214	0.447	0.280	12.112	0.430	0.202	0.378	0.280	10.821
0.15	0.346	-0.122	0.264	0.630	11.575	0.298	-0.115	0.213	0.630	10.865
0.2	0.469	0.208	0.442	0.285	11.557	0.428	0.197	0.374	0.285	10.655

0.25	0.343	-0.133	0.261	0.641	12.687	0.296	-0.126	0.210	0.641	11.250
0.3	0.508	0.204	0.340	0.878	11.522	0.437	0.185	0.279	0.878	10.952
0.35	0.630	0.464	0.460	0.532	11.587	0.561	0.427	0.390	0.532	10.661
0.4	0.507	0.203	0.340	0.879	11.431	0.437	0.184	0.279	0.879	10.597
0.45	0.453	0.172	0.414	0.320	11.451	0.410	0.162	0.347	0.320	10.583
0.5	0.629	0.462	0.459	0.535	11.452	0.560	0.426	0.389	0.535	10.573
0.55	0.337	-0.157	0.254	0.665	11.505	0.290	-0.148	0.204	0.665	10.557
0.6	0.629	0.462	0.458	0.535	11.432	0.560	0.425	0.389	0.535	10.558
0.65	0.507	0.201	0.339	0.881	11.456	0.436	0.182	0.279	0.881	10.547
0.7	0.624	0.454	0.453	0.546	11.436	0.555	0.418	0.384	0.546	10.592
0.75	0.624	0.453	0.453	0.547	11.564	0.555	0.417	0.384	0.547	10.638
0.8	0.334	-0.170	0.250	0.678	11.264	0.286	-0.161	0.201	0.678	10.407
0.85	0.334	-0.167	0.251	0.675	11.241	0.287	-0.157	0.201	0.675	10.393
0.9	0.632	0.467	0.462	0.527	11.259	0.564	0.431	0.392	0.527	10.381
0.95	0.339	-0.150	0.256	0.658	11.232	0.291	-0.142	0.205	0.658	10.361
1	0.339	-0.150	0.256	0.658	11.916	0.291	-0.141	0.206	0.658	10.916

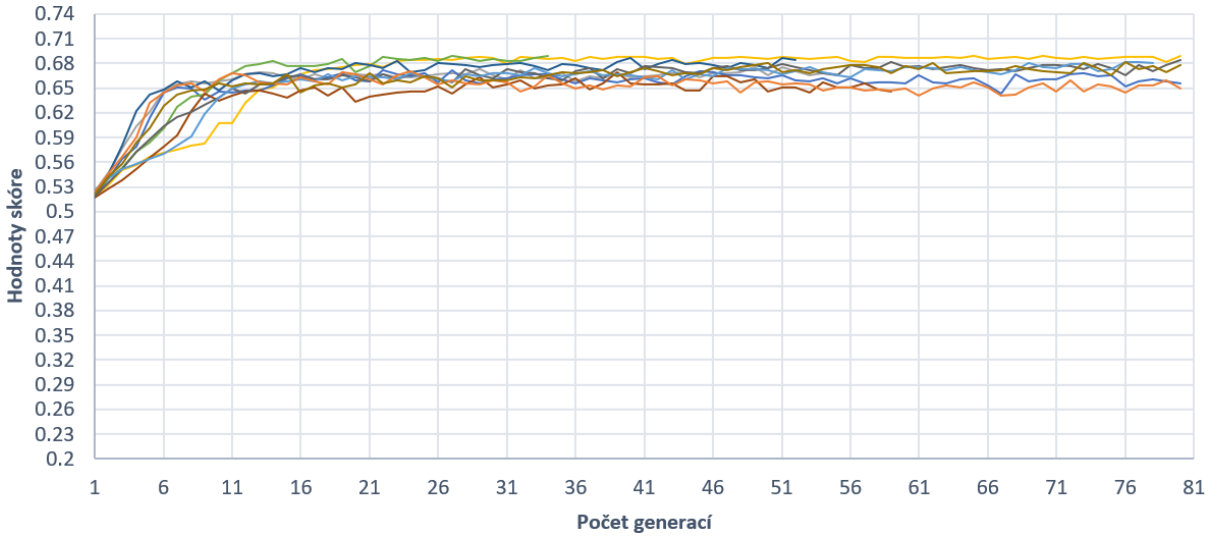
Tab. 82 – Základní srovnání algoritmu OCSVM pro různé hodnoty gamma v rámci detekce anomálií pro pátý a šestý kybernetický útok – (dataset 3). [vlastní zdroj]

	CA3_5					**CA3_6**				
Gamma	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
0.05	0.424	0.215	0.369	0.264	10.384	0.486	0.232	0.473	0.264	11.686
0.1	0.415	0.198	0.355	0.280	11.290	0.479	0.215	0.459	0.280	12.636
0.15	0.283	-0.112	0.197	0.630	10.995	0.354	-0.123	0.274	0.630	11.965
0.2	0.413	0.193	0.351	0.285	10.422	0.476	0.210	0.454	0.285	11.747
0.25	0.280	-0.123	0.194	0.641	10.598	0.351	-0.134	0.270	0.641	11.971
0.3	0.413	0.178	0.260	0.878	12.610	0.520	0.207	0.351	0.878	14.054
0.35	0.537	0.414	0.367	0.532	10.422	0.641	0.470	0.472	0.532	11.740
0.4	0.413	0.178	0.260	0.879	10.348	0.520	0.206	0.351	0.879	11.658
0.45	0.394	0.159	0.325	0.320	10.325	0.460	0.173	0.426	0.320	11.645
0.5	0.536	0.413	0.366	0.535	10.301	0.640	0.468	0.471	0.535	11.604
0.55	0.274	-0.145	0.189	0.665	10.319	0.345	-0.158	0.263	0.665	11.625
0.6	0.536	0.413	0.366	0.535	10.301	0.640	0.468	0.470	0.535	11.603
0.65	0.412	0.176	0.260	0.881	10.308	0.519	0.204	0.351	0.881	11.622
0.7	0.531	0.405	0.361	0.546	10.327	0.635	0.460	0.466	0.546	11.600
0.75	0.531	0.405	0.361	0.547	10.428	0.635	0.459	0.465	0.547	11.695
0.8	0.271	-0.157	0.186	0.678	10.178	0.342	-0.171	0.260	0.678	11.480
0.85	0.271	-0.154	0.186	0.675	10.134	0.343	-0.168	0.261	0.675	11.468
0.9	0.540	0.418	0.369	0.527	10.169	0.643	0.474	0.474	0.527	11.420
0.95	0.275	-0.138	0.190	0.658	10.171	0.347	-0.151	0.265	0.658	11.402
1	0.276	-0.138	0.190	0.658	10.442	0.347	-0.151	0.265	0.658	11.917

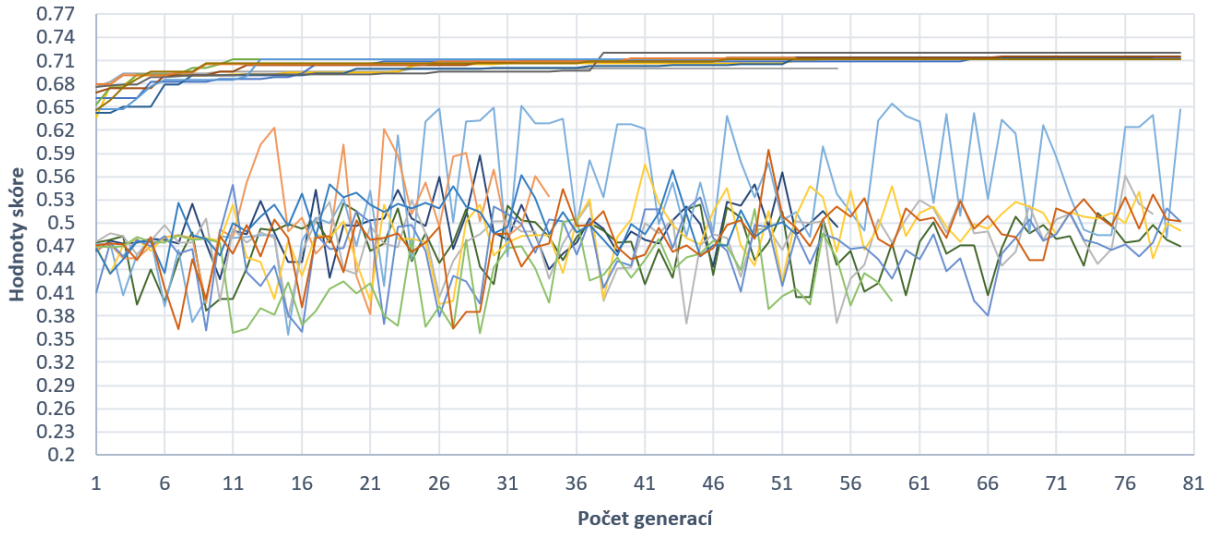
Příloha D: Optimalizace vybraných algoritmů strojového učení.

Neuronová síť – průběh optimalizace (evoluční algoritmus)

V následujících grafech jsou zobrazeny souhrnné výsledky pro zvolený algoritmus v rámci vybraných datasetů (1,2,3). Hlavním sledovaným parametrem bylo skóre vycházející z multikriteriálního hodnocení Topsis. Na Obr. 75 je zobrazen vývoj průměrného skóre pro 80 generací. Dílčí průběhy evolučního algoritmu nedosáhly 80 generací, a to především kvůli enormním časovým nárokům na výpočet. Avšak lze konstatovat, že jednotlivé evoluční algoritmy konvergují poměrně brzo. Tudíž lze předpokládat, že tyto anomálie nemají zásadní vliv na výsledná řešení. Zobrazeno je 10 oddělených průběhů tohoto optimalizačního algoritmu. Z výsledného průběhu lze vyvodit poměrně rychlou konvergenci výsledků (přibližně 11 generací) ANN do maxima a nadále průměrné hodnoty oscilují kolem hodnoty 0.66.



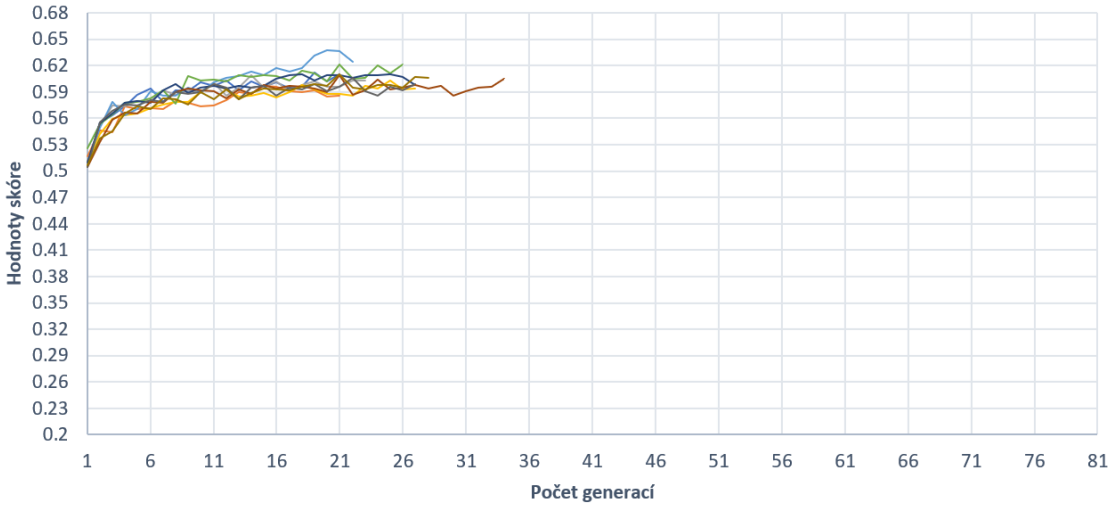
Obr. 75: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro ANN – dataset 1). [vlastní zdroj]



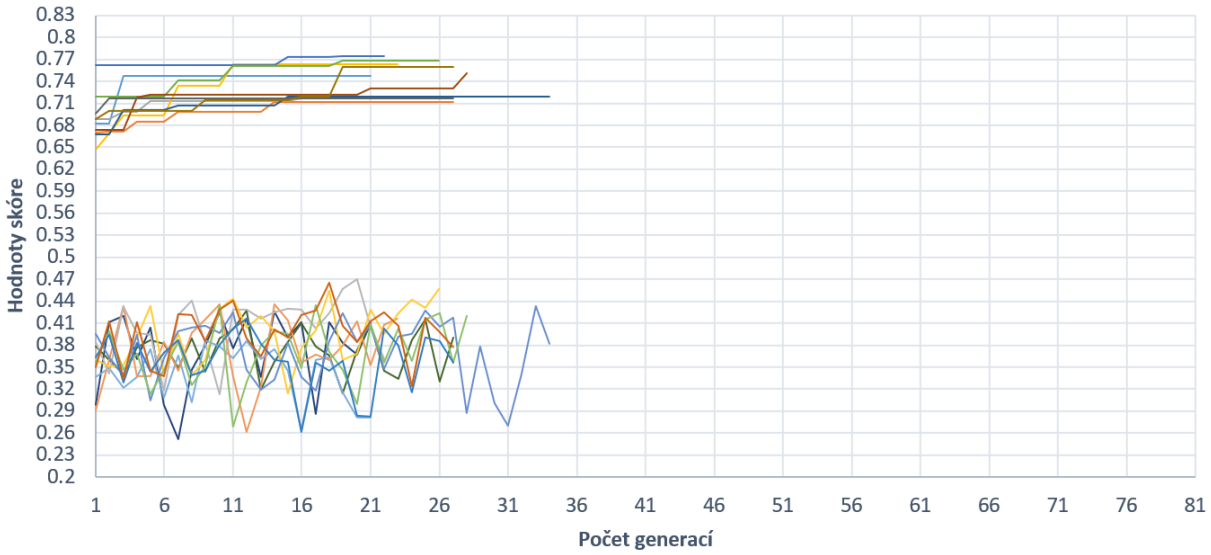
Obr. 76: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro ANN – dataset 1). [vlastní zdroj]

V případě Obr. 76 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Maximální výsledky zkonvergovaly po přibližně 11 generacích. V následujících generacích nelze pozorovat větší změny v maximálních hodnotách pro jednotlivá opakování. V případě minimální hodnoty jednotlivých generací nelze vyvodit smysluplný závěr z důvodu velké variability jednotlivých výsledků. Finální kombinace hyperparametrů, která dosáhla skóre: 0.7.

Následuje výčet výsledků, které byly získány v rámci experimentů vykonávaných na datasetu 2 (viz.Obr. 77). Zobrazeno je 10 oddělených průběhů tohoto optimalizačního algoritmu. Z výsledného průběhu lze vyvodit konvergenci výsledků (přibližně 11 generací) do maxima a nadále průměrné hodnoty oscilují kolem hodnoty 0.6.



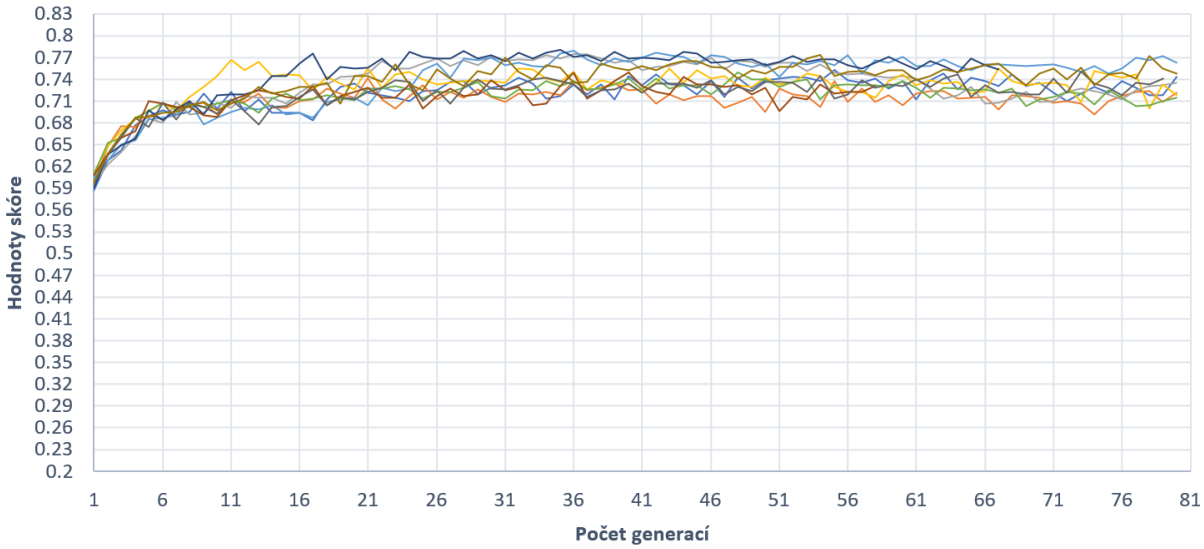
Obr. 77: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro ANN – dataset 2). [vlastní zdroj]



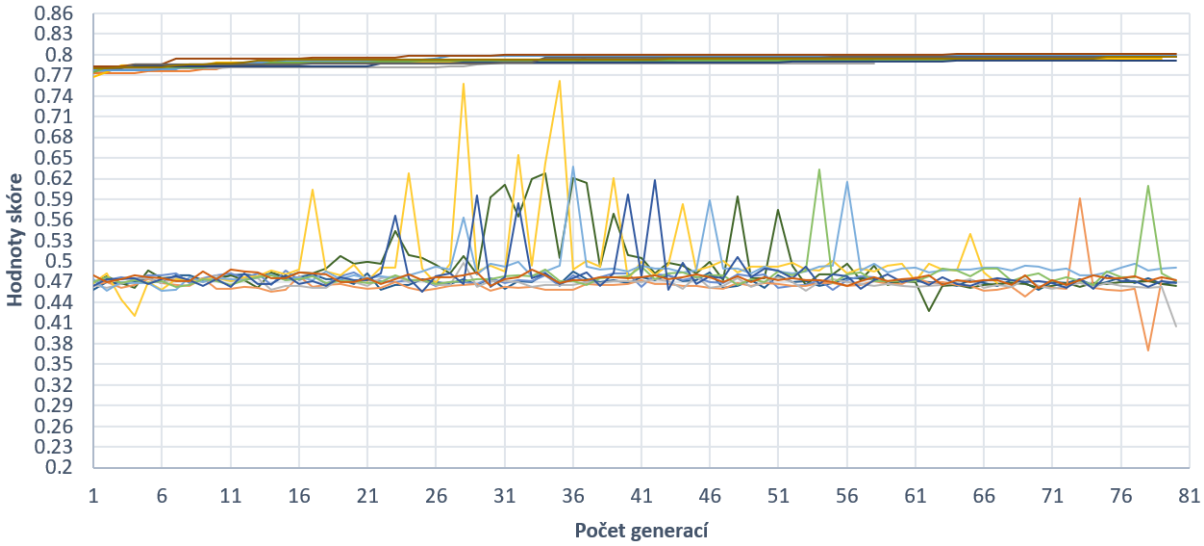
Obr. 78: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro ANN – dataset 2). [vlastní zdroj]

V případě Obr. 78 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Maximální výsledky i po přibližně 30 generacích nezkonvergovaly. V případě minimální hodnoty jednotlivých generací nelze vyvodit smysluplný závěr z důvodu velké variability jednotlivých výsledků. Finální kombinace hyperparametrů, která dosáhla skóre: 0.77.

Poslední z prezentovaných výsledků k algoritmu se vztahují k datasetu 3, kde jako v předchozích případech je realizováno deset experimentů. V těch je zobrazeno 10 oddělených průběhů optimalizačního algoritmu. V rámci Obr. 79 jsou znázorněny průběhy průměrných scóre jedinců v závislosti na generaci. Z tohoto grafu je zřejmé, že je dosaženo maximální hodnoty přibližně kolem 20. generace.



Obr. 79: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro ANN – dataset 3). [vlastní zdroj]



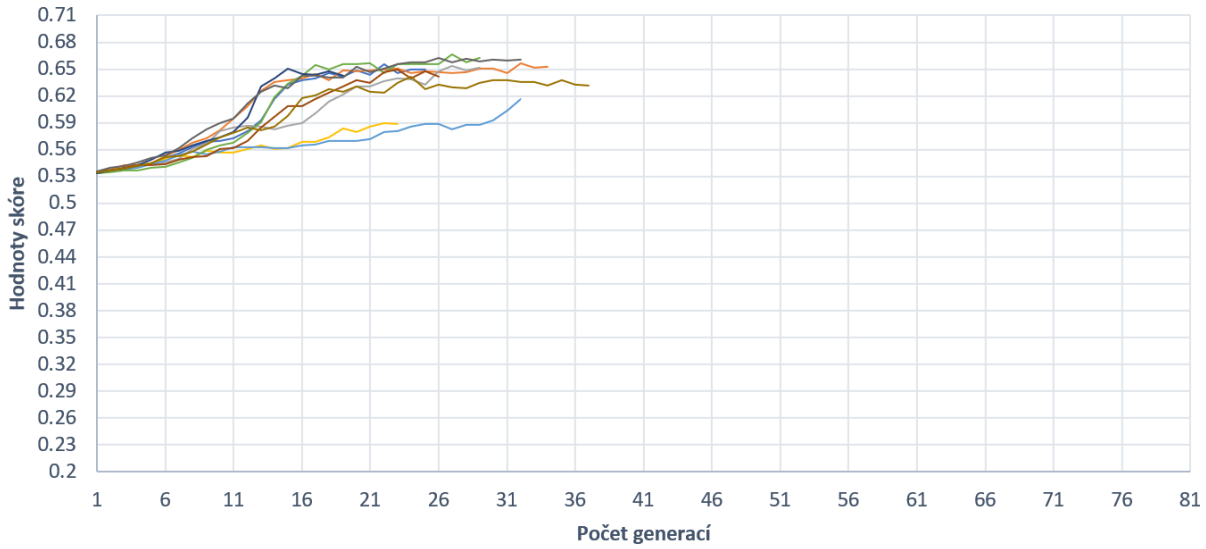
Obr. 80: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro ANN – dataset 3). [vlastní zdroj]

Obr. 80 demonstruje průběhy maximálních a minimálních hodnot jedinců v závislosti na generaci. Z průběhu lze konstatovat, že je poměrně konstantní po všechny generace. Finální kombinace hyperparametrů, která dosáhla skóre: 0.8.

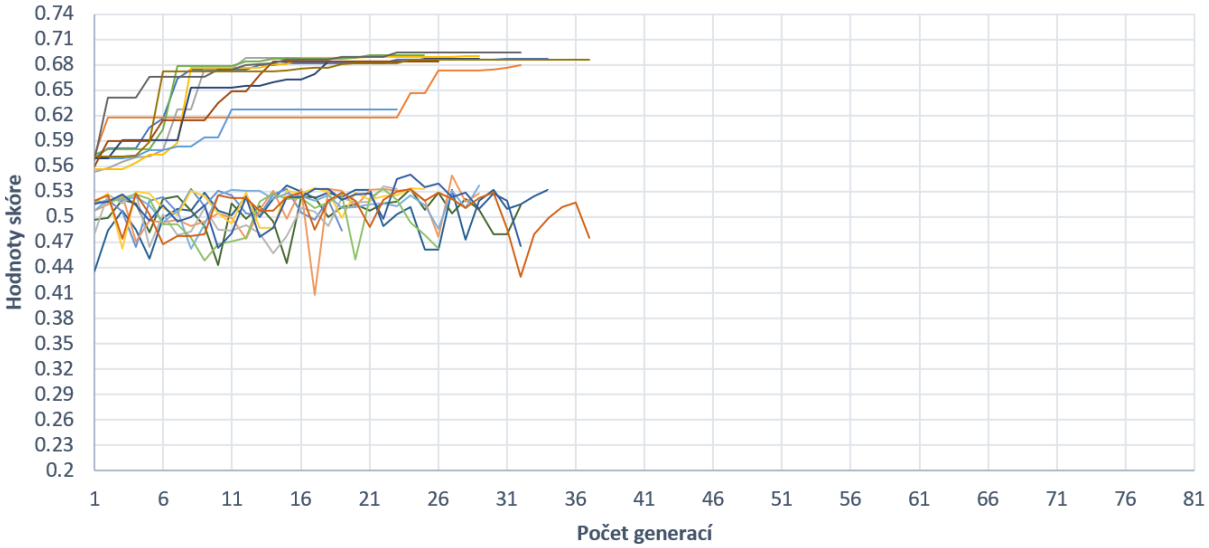
LSTM – průběh optimalizace (evoluční algoritmus)

V následujících grafech jsou zobrazeny souhrnné výsledky pro zvolený algoritmus v rámci vybraných datasetů (1,2,3). Hlavním sledovaným parametrem je skóre, vycházející z multikriteriálního hodnocení Topsis. Na Obr. 81 je zobrazen vývoj průměrného skóre pro 80 generací. Žádný ze sledovaných

průběhů evolučního algoritmu nedosáhl 80. generace. Algoritmus LSTM se z tohoto důvodu jeví jako poměrně výpočetně náročný algoritmus strojového učení. Jak je vidět z grafu na Obr. 81, žádný z 10 průběhů nedosahuje 80. generace. Většina končí kolem 30. generace. Toto poměrně brzké ukončení z pohledu generací má zásadní vliv na efektivitu získaných výsledků. Nicméně lze identifikovat hodnotu 0.62 kolem které osciluje většina z průběhů algoritmu.



Obr. 81: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro LSTM – dataset 1). [vlastní zdroj]

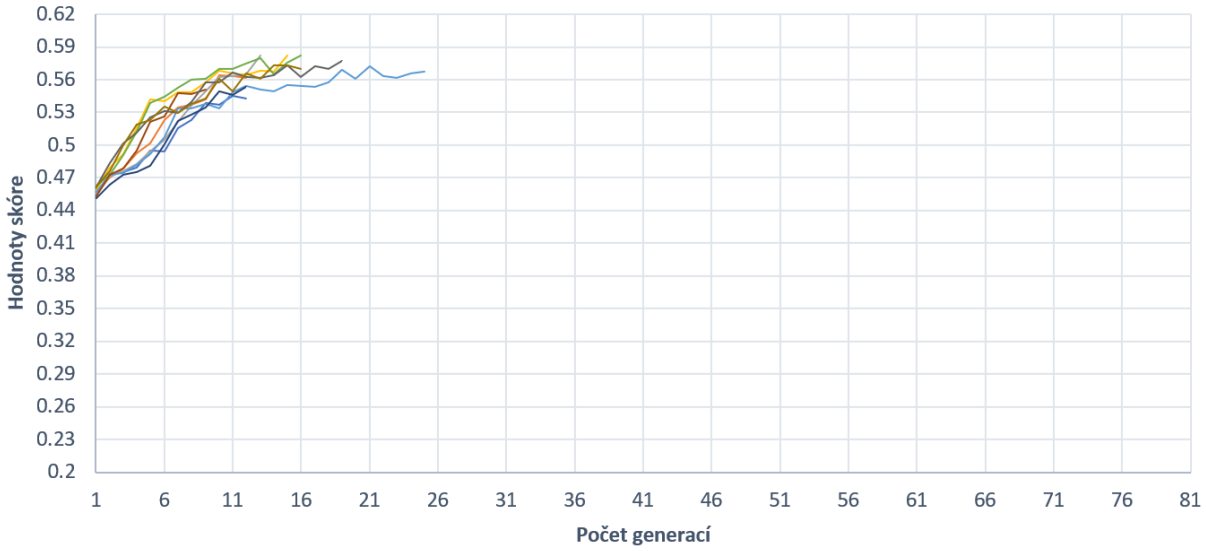


Obr. 82: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro LSTM – dataset 1). [vlastní zdroj]

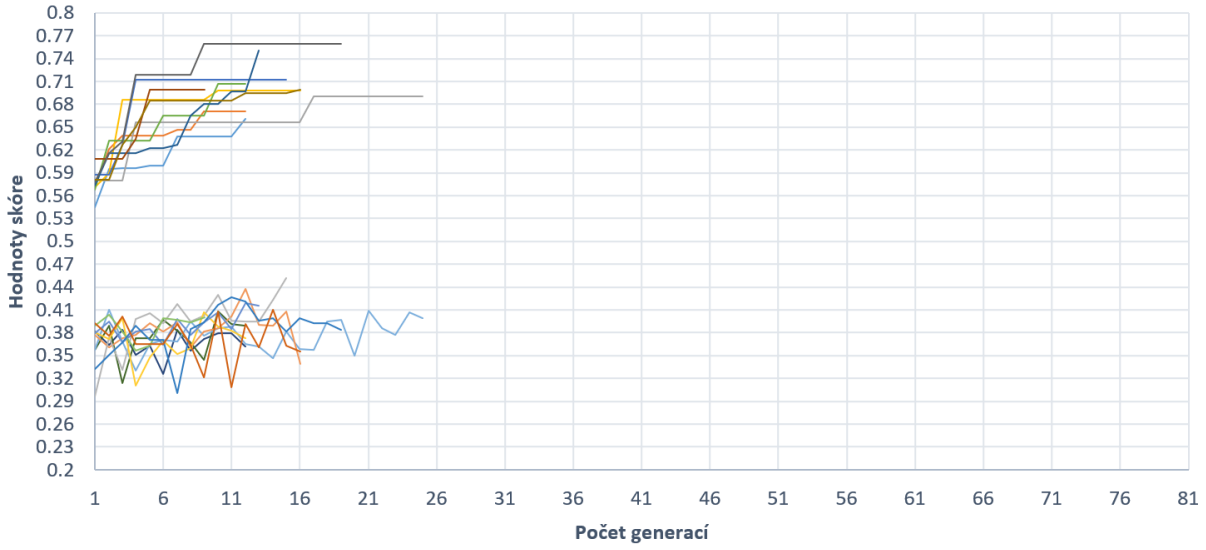
V případě Obr. 82 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé generace. Většina z maximálních výsledků zkonvergovala po přibližně 16 generacích. V případě minimální hodnoty jednotlivých generací nelze vyvodit smysluplný závěr z důvodu velké variability

jednotlivých výsledků. Je nutné poznamenat, že jednotlivé algoritmy dosáhly pouze přibližně 31 generací. Z tohoto důvodu je možné předpokládat neoptimální výsledky. Finální kombinace hyperparametrů, která dosáhla skóre: 0.68.

Následuje popis výsledků, které byly získány v rámci experimentů vykonaných na datasetu 2 (viz. Obr. 83). Zobrazeno je 10 oddělených průběhů tohoto optimalizačního algoritmu. Z výsledného průběhu lze vyvodit následující tvrzení. Počet generací v rámci vyčleněného časového intervalu byl velmi nízký z důvodů výpočetní náročností algoritmu LSTM. Z tohoto důvodu nedošlo ke konvergenci výsledků.



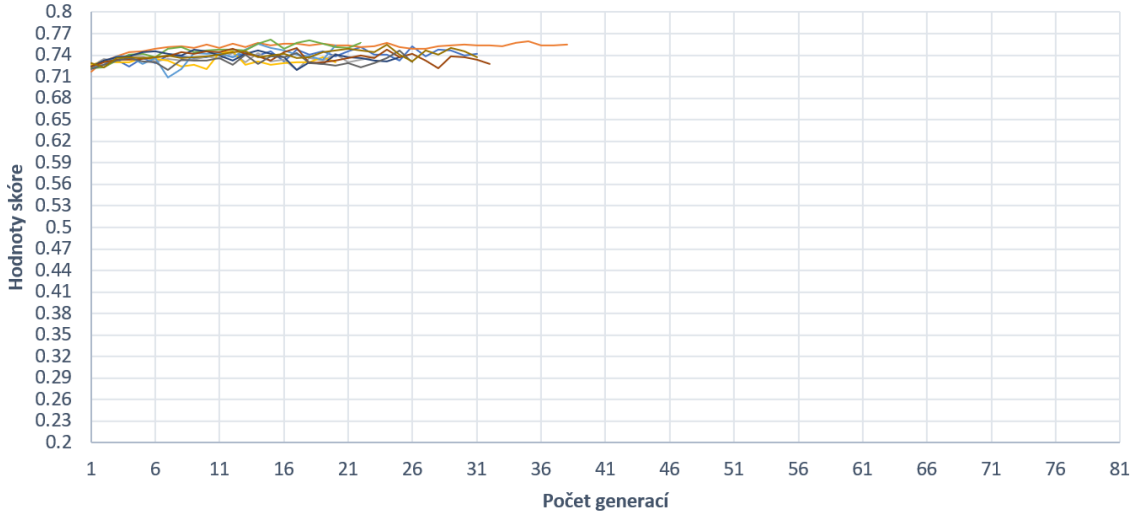
Obr. 83: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro LSTM – dataset 2). [vlastní zdroj]



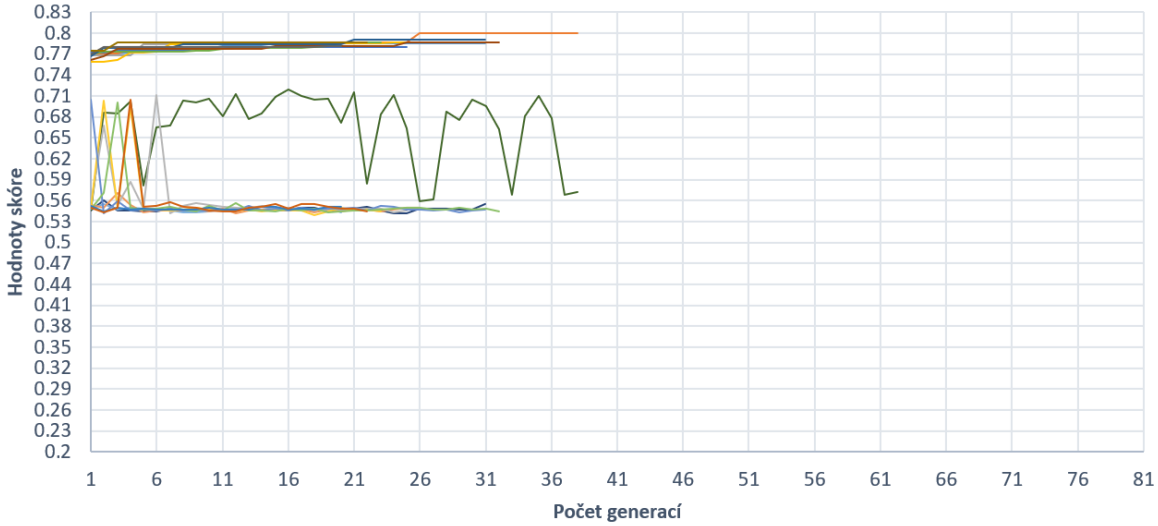
Obr. 84: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro LSTM – dataset 2). [vlastní zdroj]

V případě Obr. 84 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Z výsledků nelze vyvodit smysluplný závěr z důvodu jejich velké variability. I zde se projevují nedostatky malého počtu provedených generací. Finální kombinace hyperparametrů, která dosáhla skóre: 0.67.

Poslední experiment v rámci této sekce byl zaměřen na dataset 3. Jako v předešlých případech bylo využito LSTM algoritmu. Z deseti provedených průběhů evolučního algoritmu (viz. Obr. 85) je vidět, že jako v předešlých případech algoritmus LSTM vykazuje velké výpočetní nároky. Proto bylo provedeno přibližně 31 generací pro dílčí průběhy. Tyto průběhy vykazují jen velmi malé změny. Hodnoty jednotlivých průběhů nabývají hodnot v okolí hodnoty 0.72.



Obr. 85: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro LSTM – dataset 3). [vlastní zdroj]

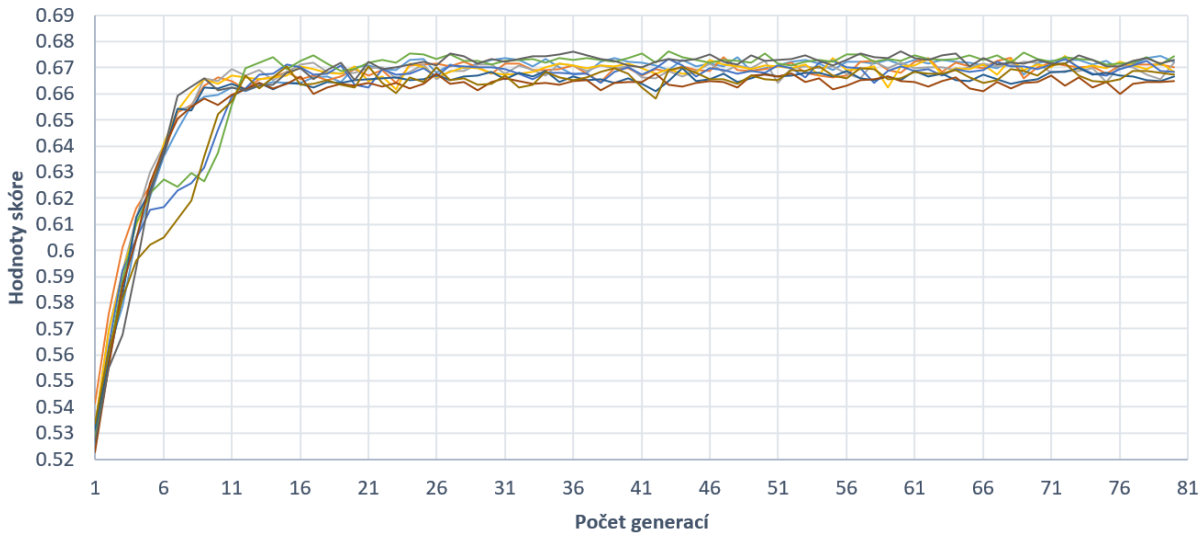


Obr. 86: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro LSTM – dataset 3). [vlastní zdroj]

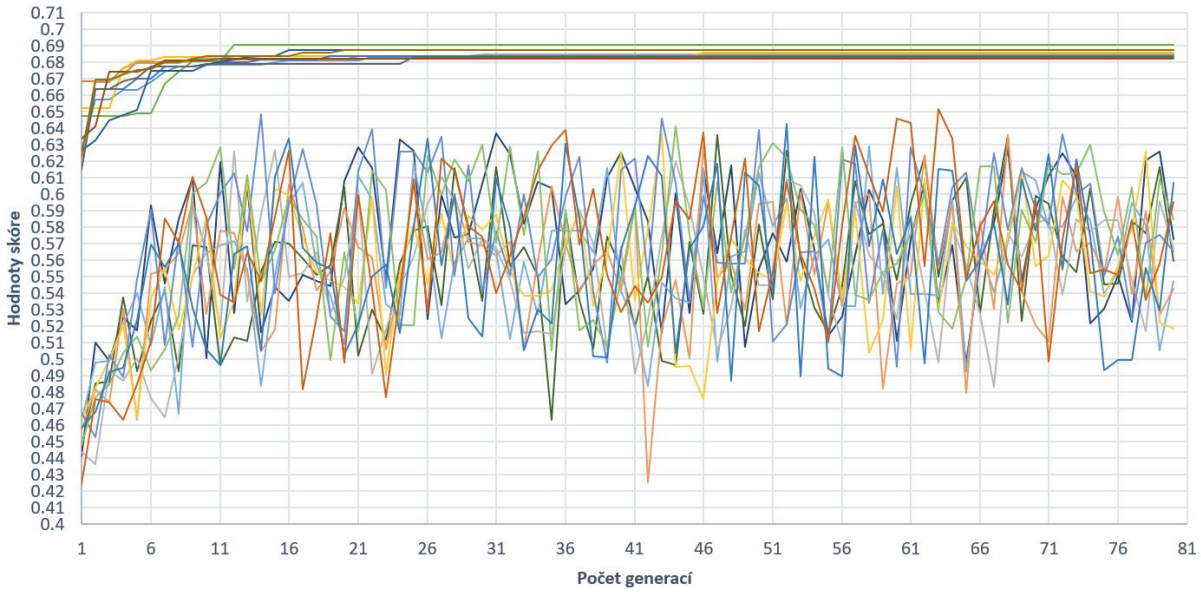
V případě Obr. 86 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Z vývoje maximální a minimální hodnoty nelze dedukovat žádné signifikantní závěry. Maximální hodnoty se ustálily v okolí hodnoty 0.77. V případě minimální hodnoty lze vidět jeden průběh, který se poměrně liší od ostatních. Finální kombinace hyperparametrů, která dosáhla skóre: 0.8.

IF – průběh optimalizace (evoluční algoritmus)

V následujících grafech jsou zobrazeny souhrnné výsledky pro zvolený algoritmus vybraných datasetů (1,2,3). Hlavním sledovaným parametrem je skóre, vycházející z multikriteriálního hodnocení Topsis. Na Obr. 87 je zobracen vývoj průměrného skóre pro 80 generací. Zobrazeno je 10 oddělených průběhů tohoto optimalizačního algoritmu. Z výsledného průběhu lze vyvodit poměrně rychlou konvergenci výsledků (přibližně 10 generací) IF do maxima a v dalším průběhu průměrné hodnoty oscilují kolem hodnoty 0.66.



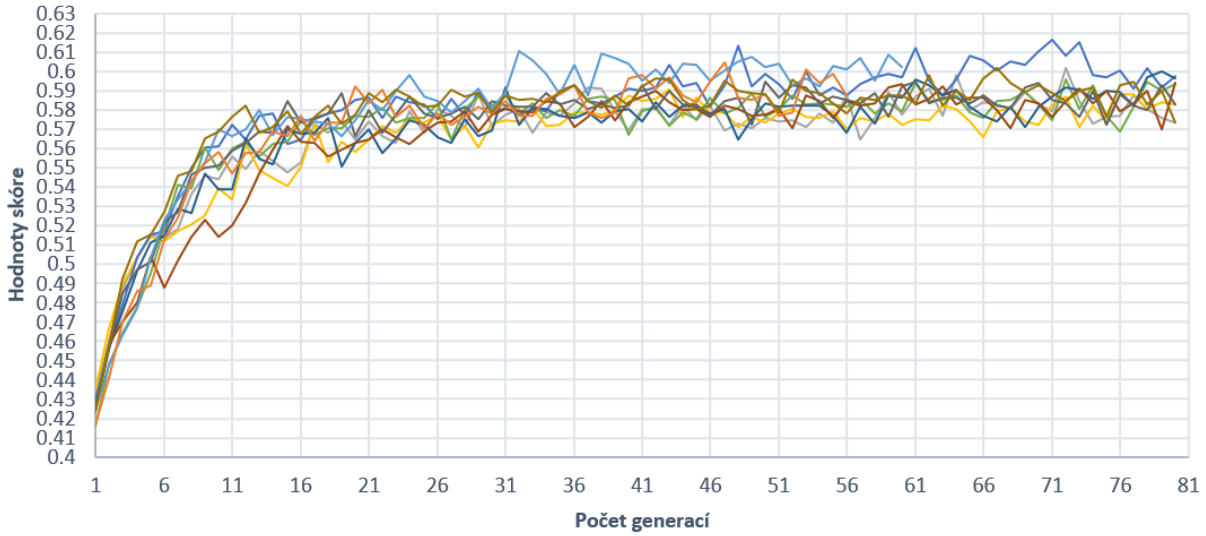
Obr. 87: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro IF – dataset 1). [vlastní zdroj]



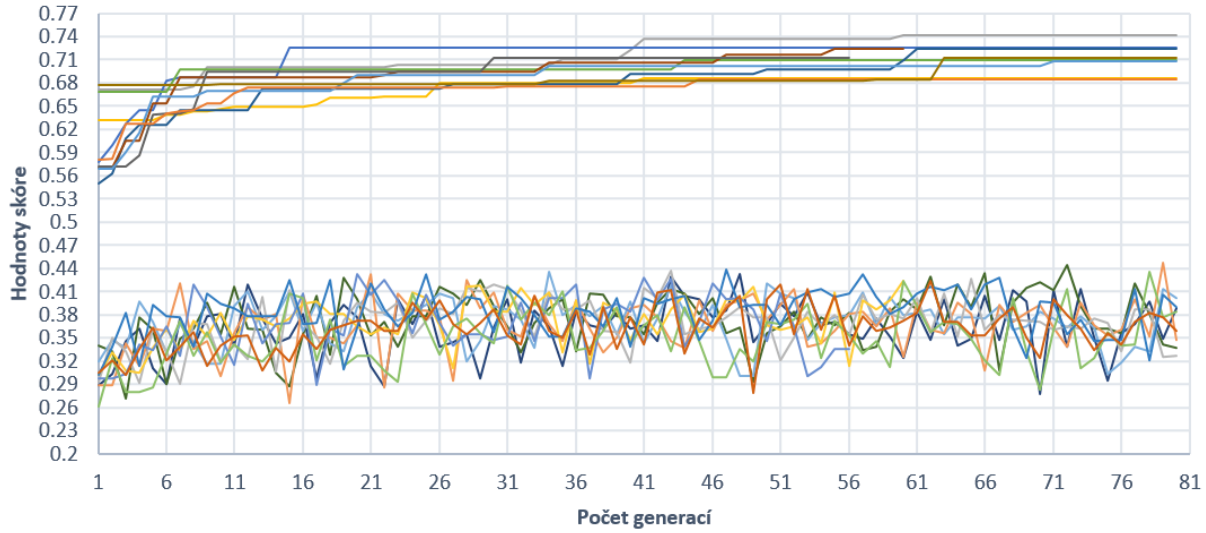
Obr. 88: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro IF – dataset 1). [vlastní zdroj]

V případě Obr. 88 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Maximální výsledky zkonvergovaly po přibližně 20 generacích. V následujících generacích nelze pozorovat větší změny v maximálních hodnotách pro jednotlivá opakování. V případě minimální hodnoty jednotlivých generací nelze vyvodit smysluplný závěr z důvodu velké variability jednotlivých výsledků. Finální kombinace hyperparametrů, která dosáhla skóre: 0.68.

V následujících odstavcích jsou popsány výsledky, které byly získány v rámci experimentů vykonaných na datasetu 2 viz. Obr. 89. Zobrazeno je 10 oddělených průběhů tohoto optimalizačního algoritmu. Z výsledného průběhu lze vyvodit poměrně rychlou konvergenci výsledků (přibližně 12 generací) IF do maxima a nadále průměrné hodnoty oscilují kolem hodnoty 0.57.



Obr. 89: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro IF – dataset 2). [vlastní zdroj]

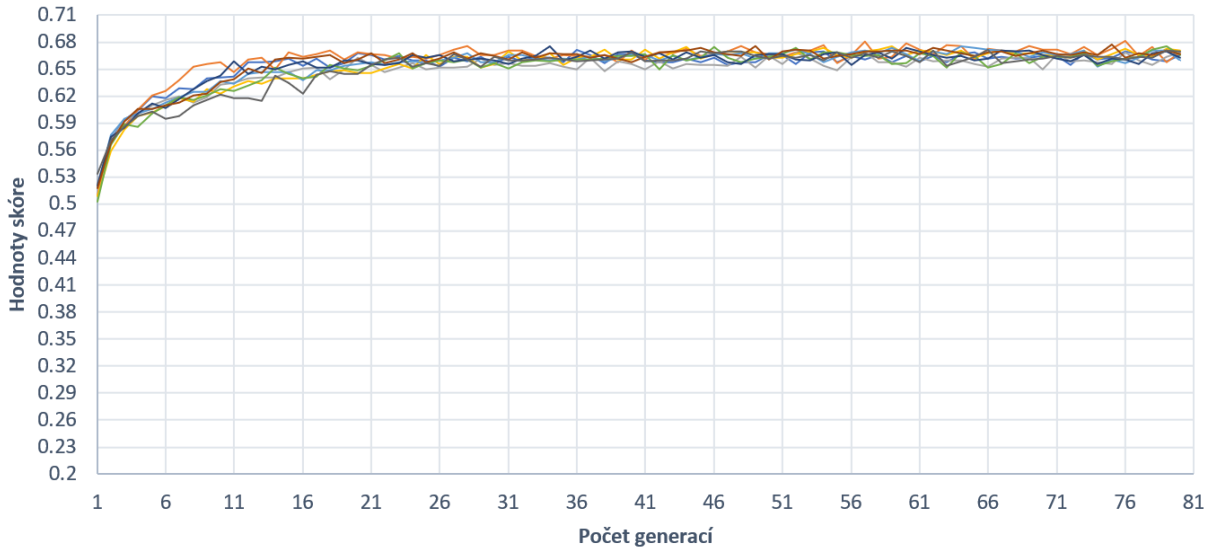


Obr. 90: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro IF – dataset 2). [vlastní zdroj]

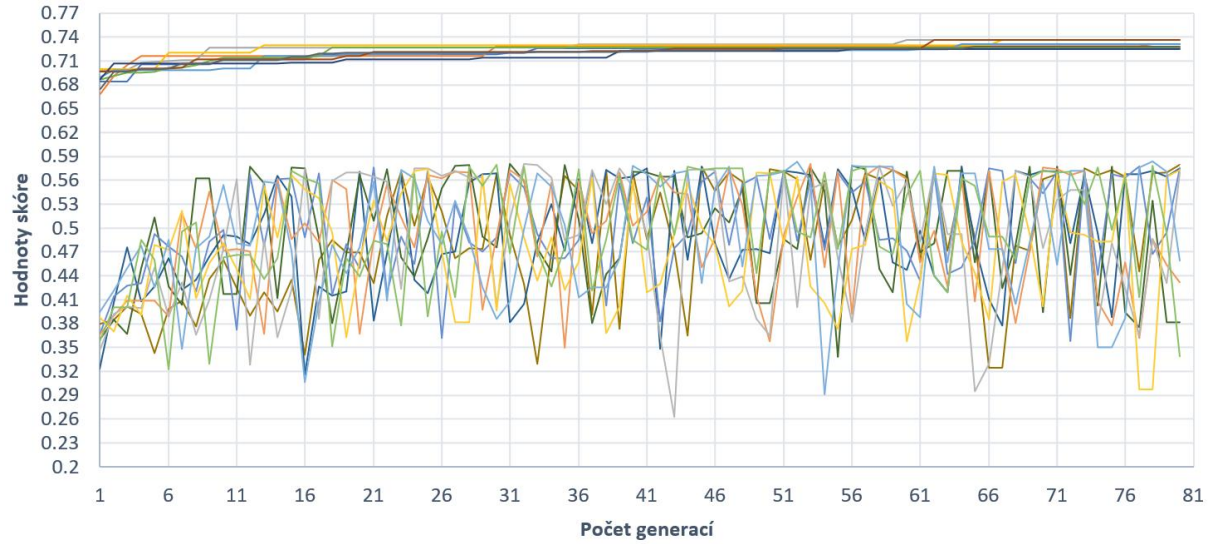
V případě Obr. 90 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Maximální výsledky zkonvergovaly po přibližně 22 generacích. V případě minimální hodnoty jednotlivých generací nelze vyvodit smysluplný závěr z důvodu velké variability jednotlivých výsledků. Finální kombinace hyperparametrů, která dosáhla skóre: 0.68.

V posledním bodu této kapitoly jsou popsány výsledky, které byly získány v díky experimentům provedeným v datasetu 3. Jako první je diskutován vývoj průměrného skóre viz. Obr. 91. Stejně jako v předešlých experimentech této kapitoly bylo provedeno deset oddělených průběhů algoritmu. Z výsledného

průběhu lze vyvodit viditelnou konvergenci výsledků (přibližně 18 generací) algoritmu IF do maxima. Přičemž zbývající průměrné hodnoty oscilují kolem hodnoty 0.65.



Obr. 91: Vývoj průměrného skóre pro jedince v rámci jednotlivých generací (pro IF – dataset 3). [vlastní zdroj]



Obr. 92: Vývoj maximálního a minimálního skóre pro jedince v rámci jednotlivých generací (pro IF – dataset 3). [vlastní zdroj]

V případě Obr. 92 jsou prezentovány výsledky pro maximální a minimální hodnotu jedince v rámci každé z generací. Maximální výsledky jedince zkonvergovaly po přibližně 11 generacích. V případě minimální hodnoty jednotlivých generací nelze vyvodit smysluplný závěr z důvodu velké variability jednotlivých výsledků. Finální kombinace hyperparametrů, která dosáhla skóre: 0.72.

Příloha E: Výsledky algoritmů neuronová síť, LSTM a IF pro finální nastavení hyperparametrů v rámci tří datasetů.

Tab. 83 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 1). [vlastní zdroj]

		CA1_1					CA1_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.843	0.840	0.961	0.003	0.051	0.840	0.666	0.952	0.070	0.016
	Max	0.975	0.973	1.000	0.020	0.065	0.993	0.980	1.000	0.466	0.020
	Min	0.725	0.702	0.769	0.000	0.046	0.720	0.236	0.734	0.000	0.014
LSTM	**Průměr**	0.278	0.212	0.292	0.066	0.593	0.671	0.150	0.737	0.468	0.034
	Max	0.904	0.895	0.937	0.113	0.746	0.960	0.884	0.980	0.983	0.051
	Min	0.000	-0.104	0.000	0.005	0.551	0.445	-0.485	0.499	0.037	0.029
Isolation forest	**Průměr**	0.426	0.388	0.314	0.152	0.183	0.782	0.610	0.986	0.017	0.205
	Max	0.709	0.683	0.724	0.228	0.254	0.808	0.644	0.989	0.026	0.379
	Min	0.315	0.264	0.208	0.025	0.166	0.756	0.577	0.979	0.014	0.186

Tab. 84 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 1). [vlastní zdroj]

		CA1_3					CA1_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.852	0.857	0.968	0.001	0.027	0.137	0.099	0.218	0.051	0.083
	Max	0.973	0.972	1.000	0.003	0.037	0.999	0.478	1.000	0.999	0.150
	Min	0.564	0.600	0.833	0.000	0.023	0.000	-0.002	0.000	0.000	0.074
LSTM	**Průměr**	0.199	0.174	0.207	0.037	0.048	0.001	-0.007	0.001	0.093	0.117
	Max	0.884	0.882	0.903	0.113	0.066	0.105	0.105	0.111	0.306	0.147
	Min	0.000	-0.054	0.000	0.002	0.043	0.000	-0.020	0.000	0.001	0.107
Isolation forest	**Průměr**	0.644	0.637	0.619	0.010	0.319	0.000	-0.004	0.000	0.016	0.812
	Max	0.719	0.712	0.705	0.016	0.428	0.009	0.018	0.005	0.025	0.965
	Min	0.456	0.444	0.456	0.007	0.293	0.000	-0.005	0.000	0.011	0.752

Tab. 85 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 2). [vlastní zdroj]

		CA2_1					CA2_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.273	-0.008	0.276	0.277	3.790	0.211	-0.012	0.215	0.220	3.755
	Max	0.362	0.117	0.369	0.309	4.273	0.282	0.076	0.284	0.248	5.620
	Min	0.192	-0.120	0.194	0.238	3.664	0.128	-0.122	0.129	0.188	3.642
LSTM	**Průměr**	0.281	0.006	0.286	0.271	6.366	0.227	0.009	0.232	0.213	5.645

	Max	0.318	0.054	0.322	0.281	7.320	0.272	0.071	0.282	0.227	6.465
	Min	0.249	-0.030	0.260	0.255	6.119	0.167	-0.063	0.173	0.183	5.397
Isolation forest	**Průměr**	0.141	0.064	0.491	0.097	7.547	0.139	-0.011	0.283	0.152	7.367
	Max	0.600	0.160	1.000	0.718	11.462	0.678	0.089	1.000	0.776	10.626
	Min	0.000	-0.077	0.000	0.000	6.250	0.000	-0.095	0.000	0.000	6.172

Tab. 86 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 2). [vlastní zdroj]

		CA2_3					**CA2_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.230	0.028	0.234	0.199	3.796	0.197	-0.002	0.201	0.195	3.788
	Max	0.328	0.154	0.338	0.224	4.580	0.249	0.062	0.253	0.207	4.279
	Min	0.149	-0.077	0.151	0.168	3.680	0.138	-0.067	0.145	0.174	3.680
LSTM	**Průměr**	0.228	0.026	0.233	0.198	5.647	0.195	-0.004	0.199	0.195	5.673
	Max	0.277	0.095	0.292	0.210	6.327	0.207	0.014	0.214	0.201	6.735
	Min	0.192	-0.021	0.195	0.172	5.417	0.173	-0.026	0.181	0.179	5.454
Isolation forest	**Průměr**	0.074	-0.010	0.198	0.070	7.095	0.092	0.000	0.207	0.079	7.328
	Max	0.695	0.110	0.905	0.788	11.556	0.708	0.091	1.000	0.798	16.379
	Min	0.000	-0.093	0.000	0.000	6.185	0.000	-0.080	0.000	0.000	6.188

Tab. 87 – Srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 2). [vlastní zdroj]

		CA2_5					**CA2_6**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.195	-0.010	0.199	0.202	3.793	0.224	-0.003	0.227	0.223	3.786
	Max	0.293	0.110	0.296	0.221	4.383	0.297	0.102	0.314	0.243	4.283
	Min	0.102	-0.116	0.109	0.171	3.669	0.138	-0.103	0.145	0.182	3.671
LSTM	**Průměr**	0.213	0.013	0.218	0.197	5.656	0.215	-0.013	0.220	0.223	5.667
	Max	0.325	0.153	0.333	0.208	6.298	0.235	0.011	0.239	0.235	6.319
	Min	0.134	-0.079	0.141	0.166	5.421	0.187	-0.046	0.193	0.199	5.458
Isolation forest	**Průměr**	0.094	0.076	0.504	0.048	7.323	0.136	0.004	0.283	0.131	7.296
	Max	0.701	0.225	1.000	0.792	13.033	0.670	0.140	1.000	0.770	10.685
	Min	0.000	-0.038	0.000	0.000	6.230	0.000	-0.119	0.000	0.000	6.195

Tab. 88 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 3). [vlastní zdroj]

		CA3_1					CA3_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.463	0.296	0.472	0.163	0.143	0.488	0.332	0.492	0.155	0.054
	Max	0.470	0.305	0.483	0.174	0.180	0.495	0.340	0.502	0.158	0.069
	Min	0.453	0.279	0.454	0.152	0.136	0.485	0.326	0.486	0.149	0.050
LSTM	**Průměr**	0.450	0.282	0.456	0.165	1.198	0.460	0.307	0.467	0.149	0.103
	Max	0.529	0.385	0.536	0.177	1.456	0.540	0.411	0.553	0.158	0.122
	Min	0.400	0.219	0.408	0.141	1.093	0.427	0.267	0.437	0.123	0.093
Isolation forest	**Průměr**	0.176	0.273	1.000	0.000	0.408	0.389	0.442	1.000	0.000	0.387
	Max	0.250	0.335	1.000	0.000	0.826	0.450	0.488	1.000	0.000	0.523
	Min	0.096	0.197	0.994	0.000	0.378	0.345	0.409	0.997	0.000	0.363

Tab. 89 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 3). [vlastní zdroj]

		CA3_3					CA3_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.425	0.082	0.412	0.385	0.059	0.470	0.298	0.486	0.163	0.055
	Max	0.561	0.252	0.527	0.688	0.070	0.490	0.321	0.501	0.185	0.071
	Min	0.000	-0.032	0.000	0.003	0.056	0.432	0.252	0.451	0.152	0.051
LSTM	**Průměr**	0.490	0.284	0.494	0.204	0.127	0.443	0.260	0.448	0.180	0.104
	Max	0.599	0.424	0.605	0.221	0.157	0.557	0.412	0.564	0.197	0.122
	Min	0.000	-0.053	0.000	0.001	0.114	0.391	0.192	0.395	0.141	0.094
Isolation forest	**Průměr**	0.124	0.209	1.000	0.000	0.423	0.631	0.000	0.551	0.739	0.405
	Max	0.258	0.327	1.000	0.000	0.583	0.638	0.000	0.557	0.746	0.752
	Min	0.016	0.075	0.985	0.000	0.392	0.000	-0.007	0.000	0.000	0.375

Tab. 90 – Srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 3). [vlastní zdroj]

		CA3_5					CA3_6				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.504	0.352	0.508	0.150	0.054	0.533	0.342	0.585	0.165	0.060
	Max	0.508	0.359	0.516	0.152	0.064	0.539	0.356	0.603	0.214	0.066
	Min	0.502	0.348	0.503	0.145	0.049	0.513	0.290	0.527	0.152	0.055
LSTM	**Průměr**	0.484	0.339	0.493	0.141	0.103	0.520	0.302	0.527	0.213	0.126
	Max	0.521	0.383	0.523	0.155	0.120	0.633	0.464	0.635	0.230	0.144
	Min	0.426	0.266	0.437	0.126	0.093	0.502	0.273	0.503	0.168	0.115
Isolation forest	**Průměr**	0.346	0.020	0.781	0.397	0.391	0.107	0.195	1.000	0.000	0.436

	Max	0.662	0.093	1.000	0.764	0.510	0.213	0.290	1.000	0.000	0.600
	Min	0.004	0.000	0.584	0.000	0.363	0.033	0.108	0.994	0.000	0.405

Tab. 91 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 1). [vlastní zdroj]

		CA1_1					**CA1_2**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.874	0.874	0.987	0.001	0.050	0.880	0.748	0.959	0.058	0.017
	Max	0.975	0.973	1.000	0.007	0.063	0.993	0.980	0.999	0.159	0.020
	Min	0.742	0.751	0.903	0.000	0.046	0.778	0.510	0.889	0.002	0.014
LSTM	**Průměr**	0.644	0.616	0.689	0.026	0.572	0.880	0.690	0.918	0.145	0.031
	Max	0.954	0.950	0.974	0.154	1.015	0.986	0.959	0.990	0.993	0.037
	Min	0.000	-0.124	0.000	0.002	0.540	0.469	-0.524	0.480	0.020	0.027
Isolation forest	**Průměr**	0.502	0.465	0.438	0.103	0.147	0.749	0.579	0.987	0.014	0.161
	Max	0.792	0.785	0.927	0.222	0.275	0.789	0.619	0.997	0.034	0.312
	Min	0.313	0.260	0.208	0.005	0.132	0.125	0.139	0.952	0.003	0.147

Tab. 92 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 1). [vlastní zdroj]

		CA1_3					**CA1_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.819	0.822	0.917	0.001	0.027	0.015	0.015	0.028	0.004	0.079
	Max	0.966	0.965	1.000	0.006	0.031	0.462	0.548	1.000	0.016	0.100
	Min	0.450	0.455	0.600	0.000	0.022	0.000	-0.004	0.000	0.000	0.073
LSTM	**Průměr**	0.600	0.591	0.628	0.011	0.044	0.004	0.001	0.005	0.020	0.104
	Max	0.946	0.945	0.959	0.108	0.058	0.125	0.128	0.167	0.265	0.134
	Min	0.000	-0.052	0.000	0.001	0.040	0.000	-0.018	0.000	0.000	0.097
Isolation forest	**Průměr**	0.377	0.369	0.417	0.012	0.255	0.000	-0.003	0.000	0.016	0.629
	Max	0.658	0.652	0.649	0.037	0.505	0.020	0.042	0.011	0.044	0.986
	Min	0.082	0.083	0.125	0.006	0.225	0.000	-0.006	0.000	0.009	0.569

Tab. 93 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 2). [vlastní zdroj]

		CA2_1					**CA2_2**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.303	0.035	0.309	0.263	4.117	0.223	0.005	0.228	0.214	4.113
	Max	0.493	0.304	0.511	0.325	5.565	0.324	0.146	0.347	0.252	6.633
	Min	0.150	-0.177	0.152	0.180	3.649	0.078	-0.179	0.080	0.165	3.613
LSTM	**Průměr**	0.263	-0.016	0.271	0.271	4.640	0.169	-0.060	0.175	0.221	4.117

	Max	0.287	0.016	0.294	0.282	6.521	0.191	-0.031	0.199	0.225	5.035
	Min	0.242	-0.044	0.250	0.255	4.389	0.160	-0.071	0.167	0.213	3.903
Isolation forest	**Průměr**	0.254	0.030	0.310	0.190	13.801	0.155	-0.054	0.176	0.190	13.951
	Max	0.312	0.122	0.400	0.256	22.398	0.214	0.024	0.248	0.252	33.636
	Min	0.190	-0.061	0.228	0.143	12.119	0.091	-0.131	0.104	0.138	11.782

Tab. 94 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 2). [vlastní zdroj]

		CA2_3					**CA2_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.223	0.019	0.227	0.199	4.246	0.253	0.067	0.258	0.182	4.207
	Max	0.379	0.214	0.383	0.227	6.585	0.370	0.214	0.377	0.204	6.969
	Min	0.098	-0.135	0.101	0.162	3.644	0.189	-0.016	0.189	0.153	3.626
LSTM	**Průměr**	0.263	0.074	0.272	0.182	4.153	0.184	-0.014	0.191	0.190	4.123
	Max	0.273	0.085	0.282	0.186	5.078	0.200	0.012	0.213	0.200	5.110
	Min	0.235	0.045	0.250	0.174	3.950	0.148	-0.059	0.153	0.176	3.930
Isolation forest	**Průměr**	0.263	0.063	0.260	0.204	13.747	0.216	0.012	0.212	0.209	13.689
	Max	0.321	0.120	0.297	0.255	25.068	0.255	0.065	0.257	0.243	19.882
	Min	0.185	-0.017	0.198	0.160	11.686	0.167	-0.045	0.169	0.171	11.780

Tab. 95 – Srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 2). [vlastní zdroj]

		CA2_5					**CA2_6**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.210	0.008	0.214	0.198	4.209	0.287	0.079	0.292	0.204	4.225
	Max	0.333	0.164	0.341	0.232	6.012	0.408	0.235	0.415	0.246	5.873
	Min	0.085	-0.150	0.086	0.165	3.647	0.175	-0.071	0.175	0.150	3.632
LSTM	**Průměr**	0.145	-0.066	0.152	0.203	4.159	0.218	0.002	0.232	0.203	4.148
	Max	0.202	0.001	0.208	0.212	5.357	0.276	0.074	0.291	0.231	5.352
	Min	0.113	-0.106	0.118	0.191	3.925	0.157	-0.082	0.164	0.187	3.932
Isolation forest	**Průměr**	0.280	0.113	0.308	0.152	13.892	0.201	-0.013	0.219	0.199	13.973
	Max	0.368	0.205	0.404	0.211	19.575	0.256	0.078	0.303	0.247	22.268
	Min	0.213	0.021	0.224	0.113	11.593	0.141	-0.097	0.151	0.151	11.609

Tab. 96 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 3). [vlastní zdroj]

		CA3_1					CA3_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.465	0.297	0.470	0.165	0.465	0.488	0.331	0.492	0.155	0.488
	Max	0.479	0.313	0.481	0.174	0.479	0.493	0.340	0.502	0.158	0.493
	Min	0.454	0.280	0.454	0.157	0.454	0.485	0.327	0.487	0.149	0.485
LSTM	**Průměr**	0.450	0.282	0.455	0.165	0.631	0.479	0.333	0.489	0.142	0.057
	Max	0.471	0.320	0.502	0.174	1.015	0.499	0.362	0.519	0.153	0.102
	Min	0.433	0.259	0.436	0.137	0.584	0.466	0.313	0.468	0.127	0.049
Isolation forest	**Průměr**	0.459	0.273	0.433	0.204	0.320	0.458	0.275	0.428	0.204	0.313
	Max	0.470	0.292	0.450	0.217	0.387	0.474	0.300	0.450	0.216	0.419
	Min	0.449	0.256	0.417	0.190	0.311	0.432	0.242	0.406	0.187	0.304

Tab. 97 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 3). [vlastní zdroj]

		CA3_3					CA3_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.410	0.058	0.378	0.403	0.410	0.453	0.273	0.463	0.175	0.453
	Max	0.561	0.287	0.516	0.688	0.561	0.483	0.311	0.491	0.185	0.483
	Min	0.000	-0.048	0.000	0.003	0.000	0.441	0.255	0.449	0.157	0.441
LSTM	**Průměr**	0.518	0.097	0.466	0.496	0.070	0.462	0.286	0.467	0.174	0.058
	Max	0.568	0.288	0.508	0.694	0.134	0.482	0.312	0.491	0.186	0.082
	Min	0.000	-0.031	0.000	0.000	0.063	0.423	0.234	0.428	0.160	0.050
Isolation forest	**Průměr**	0.509	0.296	0.524	0.203	0.338	0.434	0.236	0.426	0.203	0.319
	Max	0.520	0.315	0.543	0.215	0.423	0.461	0.274	0.454	0.216	0.370
	Min	0.501	0.281	0.510	0.188	0.327	0.407	0.199	0.399	0.189	0.310

Tab. 98 – Srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku. Nastavení hyperparametrů podle RS – (dataset 3). [vlastní zdroj]

		CA3_5					CA3_6				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.504	0.353	0.509	0.149	0.504	0.526	0.325	0.566	0.180	0.526
	Max	0.508	0.359	0.516	0.153	0.508	0.536	0.348	0.595	0.214	0.536
	Min	0.501	0.348	0.503	0.145	0.501	0.513	0.290	0.526	0.157	0.513
LSTM	**Průměr**	0.380	0.205	0.387	0.170	0.058	0.503	0.276	0.507	0.226	0.069
	Max	0.444	0.287	0.452	0.187	0.074	0.505	0.279	0.510	0.228	0.082
	Min	0.330	0.143	0.339	0.146	0.051	0.502	0.272	0.504	0.223	0.061
Isolation forest	**Průměr**	0.455	0.273	0.427	0.203	0.312	0.514	0.297	0.536	0.203	0.344

	Max	0.471	0.296	0.449	0.216	0.347	0.522	0.315	0.554	0.215	0.380
	Min	0.435	0.248	0.407	0.186	0.304	0.507	0.283	0.522	0.189	0.334

Tab. 99 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 1). [vlastní zdroj]

		CA1_1					**CA1_2**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.941	0.937	0.982	0.002	0.057	0.916	0.833	0.990	0.014	0.017
	Max	0.975	0.973	1.000	0.019	0.081	0.999	0.996	0.999	0.204	0.038
	Min	0.757	0.740	0.790	0.000	0.051	0.773	0.480	0.863	0.002	0.013
LSTM	**Průměr**	0.470	0.420	0.489	0.050	0.790	0.796	0.426	0.822	0.338	0.047
	Max	0.958	0.954	0.974	0.153	0.976	0.991	0.973	0.993	0.967	0.084
	Min	0.000	-0.124	0.000	0.002	0.736	0.489	-0.471	0.499	0.013	0.039
Isolation forest	**Průměr**	0.443	0.405	0.337	0.144	0.175	0.775	0.605	0.985	0.017	0.209
	Max	0.709	0.683	0.724	0.228	0.287	0.807	0.641	0.989	0.034	0.507
	Min	0.328	0.284	0.216	0.025	0.143	0.178	0.160	0.929	0.014	0.175

Tab. 100 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 1). [vlastní zdroj]

		CA1_3					**CA1_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.947	0.948	0.996	0.000	0.026	0.195	0.208	0.354	0.011	0.086
	Max	0.980	0.979	1.000	0.002	0.033	0.999	0.774	1.000	0.999	0.344
	Min	0.827	0.831	0.929	0.000	0.023	0.000	-0.002	0.000	0.000	0.074
LSTM	**Průměr**	0.423	0.402	0.443	0.037	0.068	0.043	0.038	0.054	0.077	0.166
	Max	0.932	0.930	0.958	0.148	0.084	0.444	0.447	1.000	0.238	0.215
	Min	0.000	-0.063	0.000	0.001	0.060	0.000	-0.017	0.000	0.000	0.146
Isolation forest	**Průměr**	0.601	0.594	0.581	0.011	0.360	0.001	-0.002	0.000	0.016	0.974
	Max	0.714	0.708	0.697	0.029	0.597	0.018	0.039	0.009	0.024	1.609
	Min	0.197	0.180	0.209	0.007	0.301	0.000	-0.005	0.000	0.011	0.840

Tab. 101 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 2). [vlastní zdroj]

		CA2_1					**CA2_2**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.279	0.000	0.282	0.275	3.065	0.220	-0.002	0.223	0.219	3.056
	Max	0.384	0.147	0.390	0.317	4.325	0.327	0.141	0.339	0.242	3.857
	Min	0.180	-0.139	0.181	0.228	2.875	0.130	-0.114	0.133	0.179	2.861
LSTM	**Průměr**	0.219	-0.074	0.226	0.286	7.927	0.255	0.052	0.267	0.193	6.954

	Max	0.259	-0.023	0.265	0.293	9.709	0.268	0.067	0.279	0.213	8.631
	Min	0.189	-0.113	0.197	0.270	7.468	0.212	-0.007	0.218	0.184	6.537
Isolation forest	**Průměr**	0.265	0.057	0.336	0.171	28.582	0.128	-0.073	0.154	0.174	28.811
	Max	0.327	0.150	0.429	0.219	33.919	0.185	-0.013	0.212	0.214	35.906
	Min	0.213	-0.018	0.266	0.136	25.630	0.079	-0.129	0.103	0.125	26.191

Tab. 102 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 2). [vlastní zdroj]

		CA2_3					**CA2_4**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.217	0.010	0.220	0.205	3.098	0.198	-0.002	0.201	0.197	3.113
	Max	0.294	0.106	0.296	0.227	4.156	0.274	0.093	0.277	0.217	4.058
	Min	0.124	-0.106	0.126	0.186	2.874	0.126	-0.093	0.127	0.168	2.905
LSTM	**Průměr**	0.241	0.046	0.249	0.189	6.963	0.233	0.046	0.241	0.180	6.957
	Max	0.270	0.085	0.282	0.196	8.178	0.284	0.113	0.298	0.203	8.460
	Min	0.230	0.031	0.237	0.175	6.557	0.146	-0.063	0.150	0.162	6.562
Isolation forest	**Průměr**	0.236	0.041	0.245	0.188	28.754	0.213	0.020	0.219	0.188	28.770
	Max	0.315	0.115	0.294	0.219	36.685	0.246	0.064	0.257	0.225	33.782
	Min	0.177	-0.018	0.196	0.154	25.970	0.151	-0.043	0.165	0.155	26.077

Tab. 103 – Srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 2). [vlastní zdroj]

		CA2_5					**CA2_6**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.209	0.006	0.212	0.201	3.132	0.223	-0.005	0.226	0.225	3.100
	Max	0.323	0.147	0.325	0.238	4.080	0.318	0.132	0.341	0.252	3.977
	Min	0.077	-0.162	0.078	0.174	2.890	0.141	-0.113	0.142	0.171	2.904
LSTM	**Průměr**	0.176	-0.029	0.184	0.198	6.959	0.287	0.089	0.304	0.187	6.981
	Max	0.206	0.013	0.219	0.205	8.683	0.356	0.185	0.388	0.225	9.916
	Min	0.148	-0.064	0.155	0.181	6.556	0.173	-0.055	0.185	0.155	6.566
Isolation forest	**Průměr**	0.259	0.098	0.298	0.141	28.825	0.202	-0.001	0.230	0.182	28.777
	Max	0.340	0.178	0.365	0.174	33.707	0.238	0.072	0.302	0.226	33.446
	Min	0.207	0.039	0.245	0.113	26.057	0.148	-0.080	0.162	0.136	26.105

Tab. 104 – Srovnání algoritmů pro detekci anomálií v rámci prvního a druhého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 3). [vlastní zdroj]

		CA3_1					CA3_2				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.462	0.293	0.468	0.165	0.129	0.490	0.333	0.494	0.154	0.054
	Max	0.477	0.311	0.482	0.174	0.208	0.499	0.346	0.505	0.158	0.089
	Min	0.452	0.277	0.453	0.152	0.118	0.485	0.326	0.486	0.149	0.047
LSTM	**Průměr**	0.470	0.309	0.476	0.159	0.892	0.474	0.325	0.480	0.147	0.081
	Max	0.505	0.357	0.518	0.169	1.859	0.509	0.371	0.521	0.161	0.115
	Min	0.436	0.264	0.442	0.142	0.820	0.424	0.261	0.429	0.132	0.072
Isolation forest	**Průměr**	0.169	0.266	1.000	0.000	0.383	0.385	0.439	1.000	0.000	0.363
	Max	0.283	0.361	1.000	0.000	0.786	0.439	0.480	1.000	0.000	0.534
	Min	0.087	0.186	0.993	0.000	0.307	0.345	0.409	0.997	0.000	0.294

Tab. 105 – Srovnání algoritmů pro detekci anomálií v rámci třetího a čtvrtého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 3). [vlastní zdroj]

		CA3_3					CA3_4				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.302	0.089	0.312	0.214	0.059	0.468	0.292	0.477	0.172	0.056
	Max	0.561	0.285	0.519	0.688	0.079	0.491	0.323	0.502	0.185	0.094
	Min	0.000	-0.045	0.000	0.003	0.053	0.431	0.251	0.448	0.145	0.049
LSTM	**Průměr**	0.506	0.289	0.508	0.216	0.099	0.463	0.286	0.466	0.175	0.082
	Max	0.511	0.294	0.511	0.218	0.119	0.503	0.341	0.510	0.186	0.098
	Min	0.502	0.282	0.503	0.214	0.089	0.432	0.245	0.435	0.158	0.072
Isolation forest	**Průměr**	0.099	0.184	1.000	0.000	0.387	0.625	0.000	0.546	0.731	0.370
	Max	0.280	0.343	1.000	0.000	0.541	0.638	0.000	0.557	0.746	0.515
	Min	0.012	0.063	0.984	0.000	0.310	0.000	-0.007	0.000	0.000	0.302

Tab. 106 – Srovnání algoritmů pro detekci anomálií v rámci pátého a šestého kybernetického útoku. Nastavení hyperparametrů podle TPE – (dataset 3). [vlastní zdroj]

		CA3_5					CA3_6				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.503	0.351	0.507	0.150	0.054	0.527	0.327	0.568	0.178	0.060
	Max	0.508	0.359	0.516	0.153	0.081	0.538	0.356	0.603	0.214	0.077
	Min	0.501	0.347	0.502	0.145	0.047	0.513	0.290	0.527	0.152	0.054
LSTM	**Průměr**	0.498	0.356	0.505	0.139	0.081	0.522	0.304	0.526	0.216	0.099
	Max	0.514	0.378	0.528	0.147	0.096	0.533	0.317	0.533	0.228	0.121
	Min	0.479	0.332	0.488	0.129	0.071	0.504	0.275	0.505	0.214	0.089
Isolation forest	**Průměr**	0.373	0.018	0.758	0.428	0.365	0.103	0.188	1.000	0.000	0.400

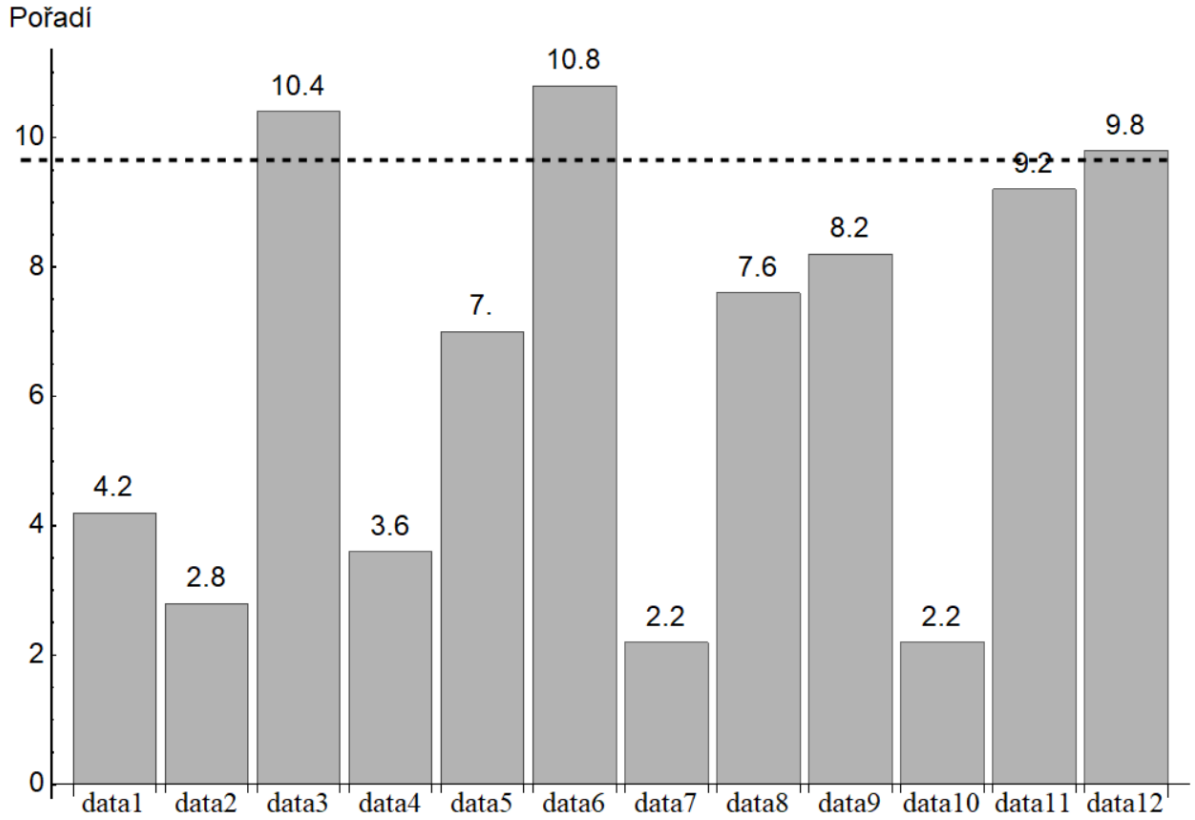
Max	0.662	0.060	1.000	0.764	0.478	0.262	0.328	1.000	0.000	0.589
Min	0.004	0.000	0.584	0.000	0.290	0.001	0.018	0.995	0.000	0.327

Friedmanův test pro jednotlivé algoritmy – dataset 1

Tab. 107 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_1 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	21.53571	3.31668×10^{-14}

Podle Tab. 107 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 % procent, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 93: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA1_1. [vlastní zdroj]

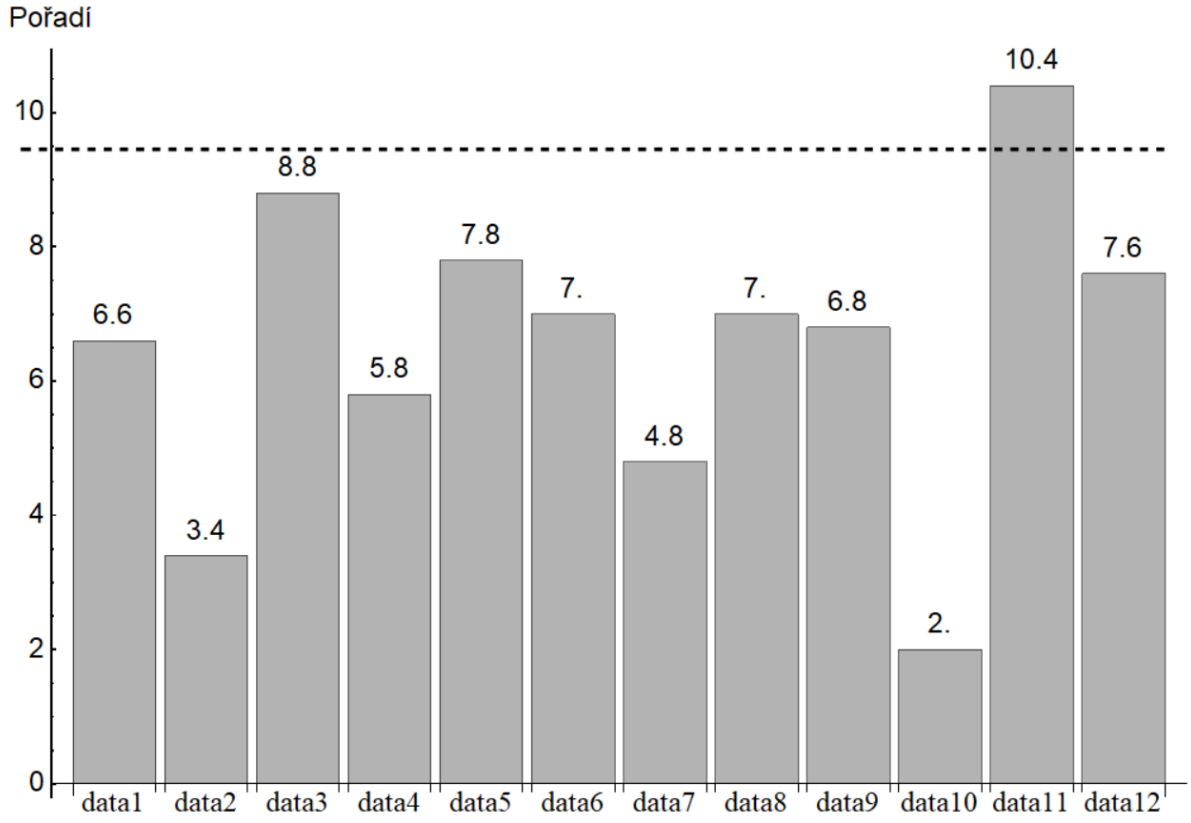
Na Obr. 93 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení IF má podstatně horší výsledky při standartním nastavení, při nastavení za pomocí evolučního algoritmu a nastavení pomocí optimalizačního algoritmu TPE (data3, data6, data12) oproti ostatním nastavením algoritmů. Také

lze konstatovat vyšší detekční schopnosti v případech neuronové sítě v podstatě každém případě. Především je-li využit některý z optimalizačních algoritmů (data4, data7, data10). Dobrého výsledku dosahuje i algoritmus LSTM v základním nastavení (data2).

Tab. 108 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_2 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.66978	0.01025995

Podle Tab. 108 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 94: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA1_2. [vlastní zdroj]

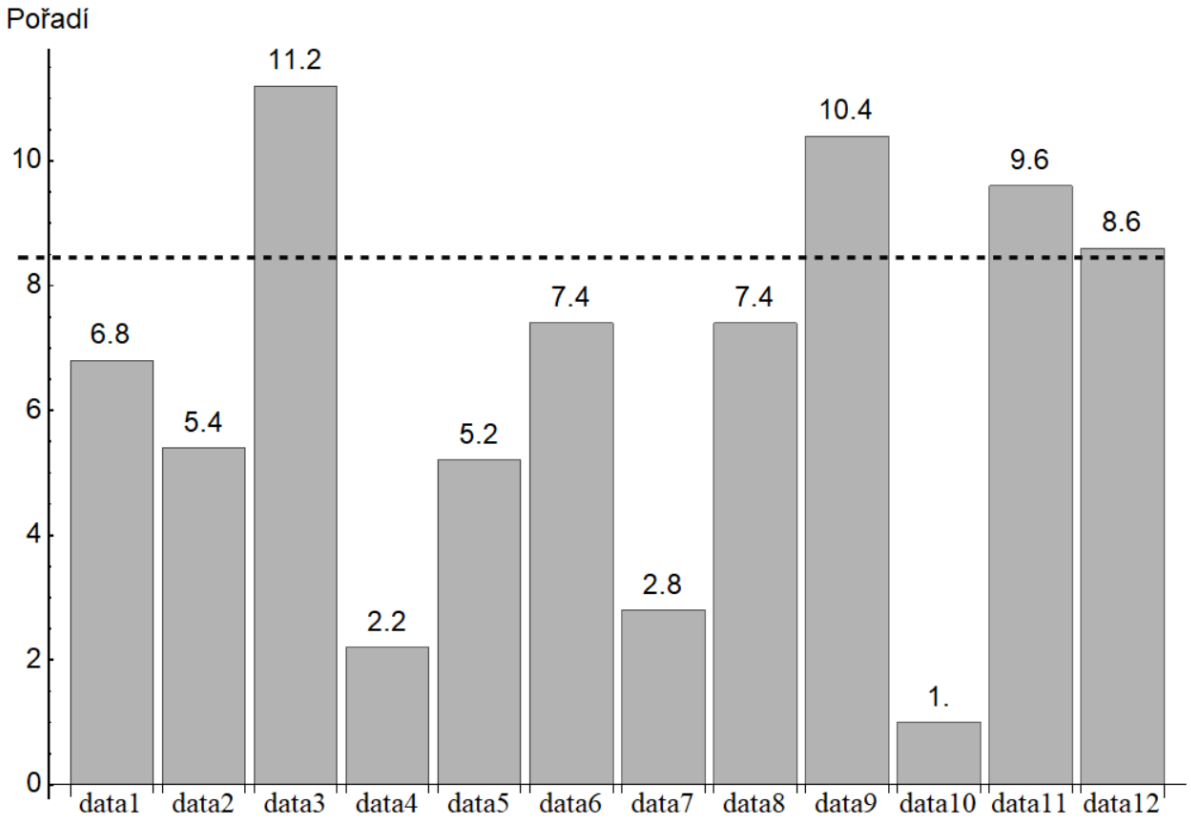
Na Obr. 94 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení LSTM má podstatně horší výsledky při nastavení podle TPE optimalizačního algoritmu (data11) oproti ostatním nastavením algoritmů. Také

lze konstatovat vyšší detekční schopnosti v případech neuronové sítě při nastavení podle TPE (data10) a LSTM při základním nastavení (data2).

Tab. 109 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_3 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	19.21429	2.49256×10^{-13}

Podle Tab. 109 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 95: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA1_3. [vlastní zdroj]

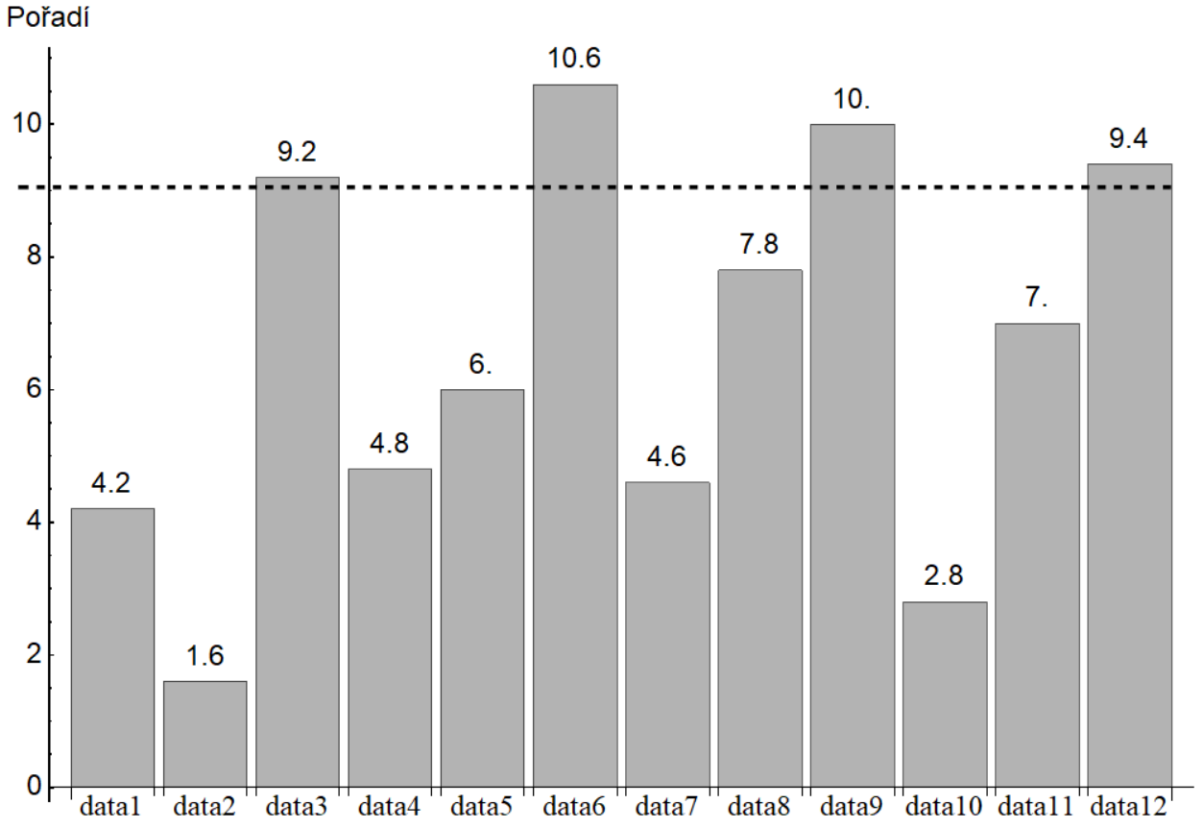
Na Obr. 95 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení IF má podstatně horší výsledky při standartním nastavení, za pomocí optimalizačního algoritmu RS a nastavení pomocí algoritmu TPE (data3, data9, data12) oproti ostatním nastavením algoritmů. Ve stejném smyslu lze označit algoritmus LSTM nastavený pomocí TPE jako odchylku od ostatních výsledků. Lze konstatovat vyšší detekční schopnosti v případech neuronové sítě

v každém případě, když byl využit optimalizační algoritmus. Jedná se o případy (data4, data7, data10).

Tab. 110 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA1_4 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	8.28522	1.26086×10^{-7}

Podle Tab. 110 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 96: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA1_4. [vlastní zdroj]

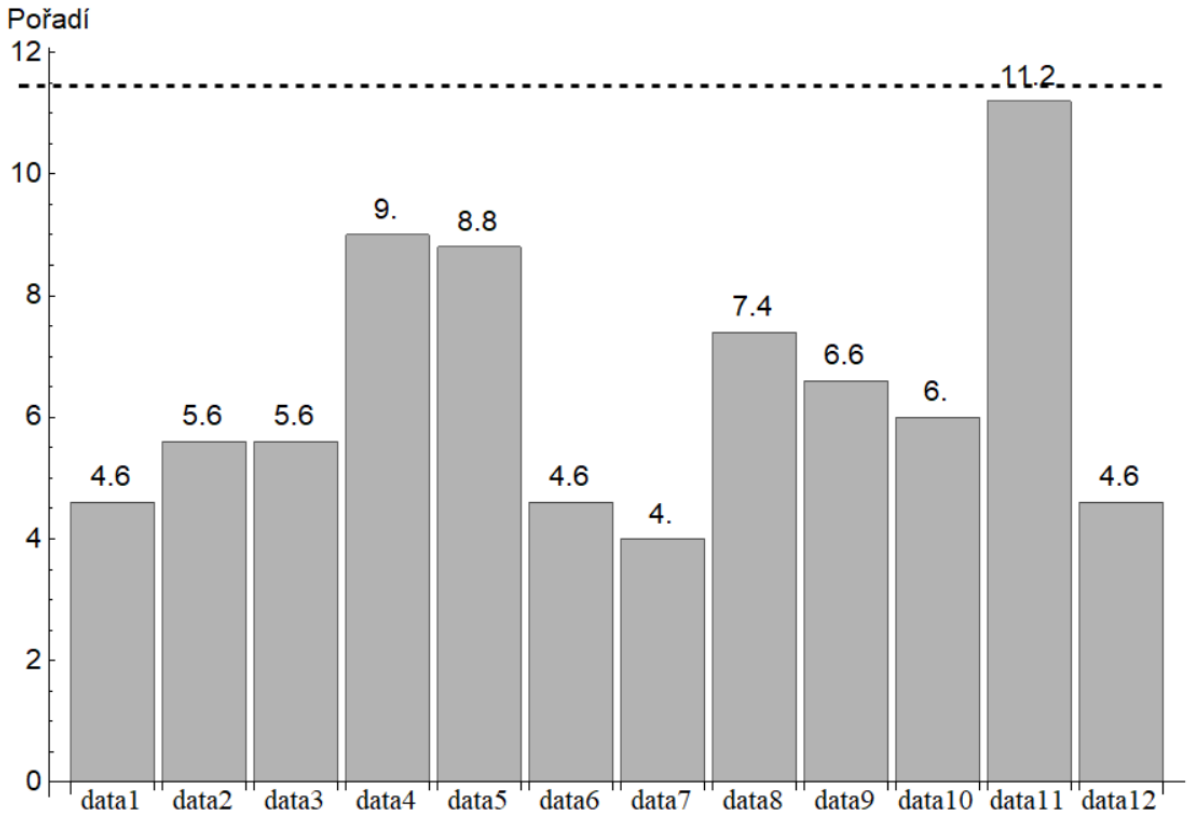
Na Obr. 96 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení IF má podstatně horší výsledky při každém nastavení (data3, data6, data9, data12) oproti ostatním nastavením algoritmů. Také lze konstatovat vyšší detekční schopnosti v případech algoritmu LSTM (data2) ve standardním nastavení a neuronové sítě téměř v každém případě (data1, data4, data7, data10).

Friedmanův test pro jednotlivé algoritmy – dataset 2

Tab. 111 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_1 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.38393	0.02048

Podle Tab. 111 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



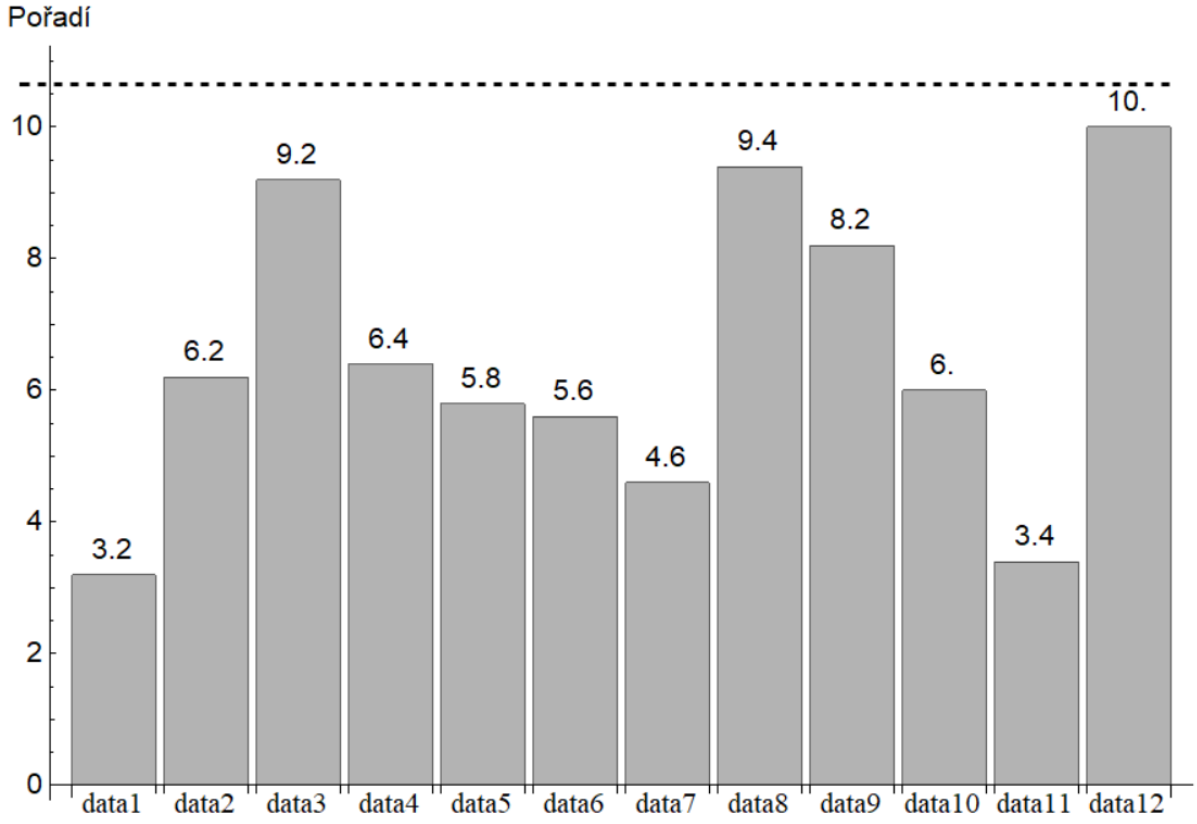
Obr. 97: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_1. [vlastní zdroj]

Na Obr. 97 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku neexistuje algoritmus strojového učení, který by se významně negativně odlišoval od ostatních algoritmů. Nejblíže k tomu má algoritmus LSTM nastavený pomocí TPE. Oproti tomu lze identifikovat standardně nastavenou neuronovou síť (data1), IF nastavený podle evolučního algoritmu (data6), neuronovou síť nastavenou pomocí RS (data7) a IF nastavený podle TPE (data12) jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 112 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_2 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.62037	0.01156

Podle Tab. 112 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



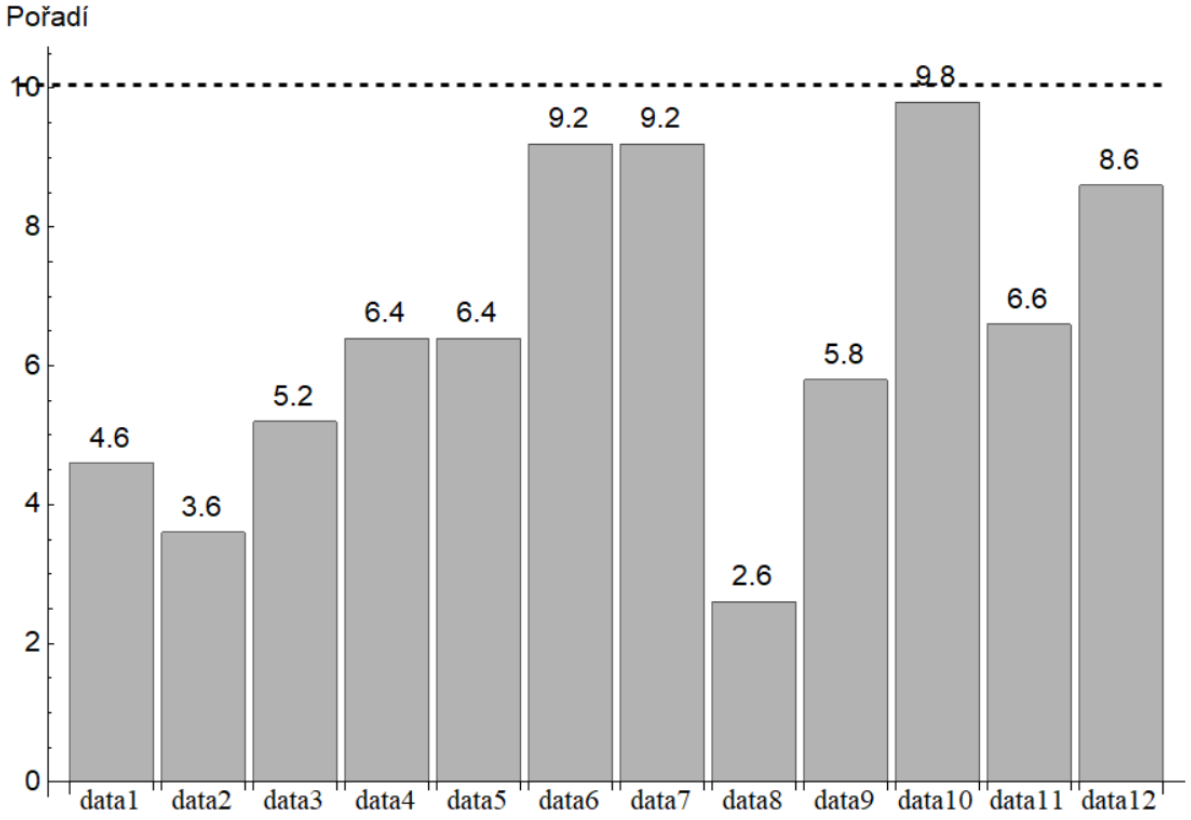
Obr. 98: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_2. [vlastní zdroj]

Na Obr. 98 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku neexistuje varianta algoritmu strojového učení, která by se významně negativně odlišovala od ostatních algoritmů. Oproti tomu, lze identifikovat standardně nastavenou neuronovou síť(data1), neuronovou síť nastavenou podle RS (data7) a LSTM nastavený podle TPE (data11) jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 113 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_3 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.83556	0.00689

Podle Tab. 113 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



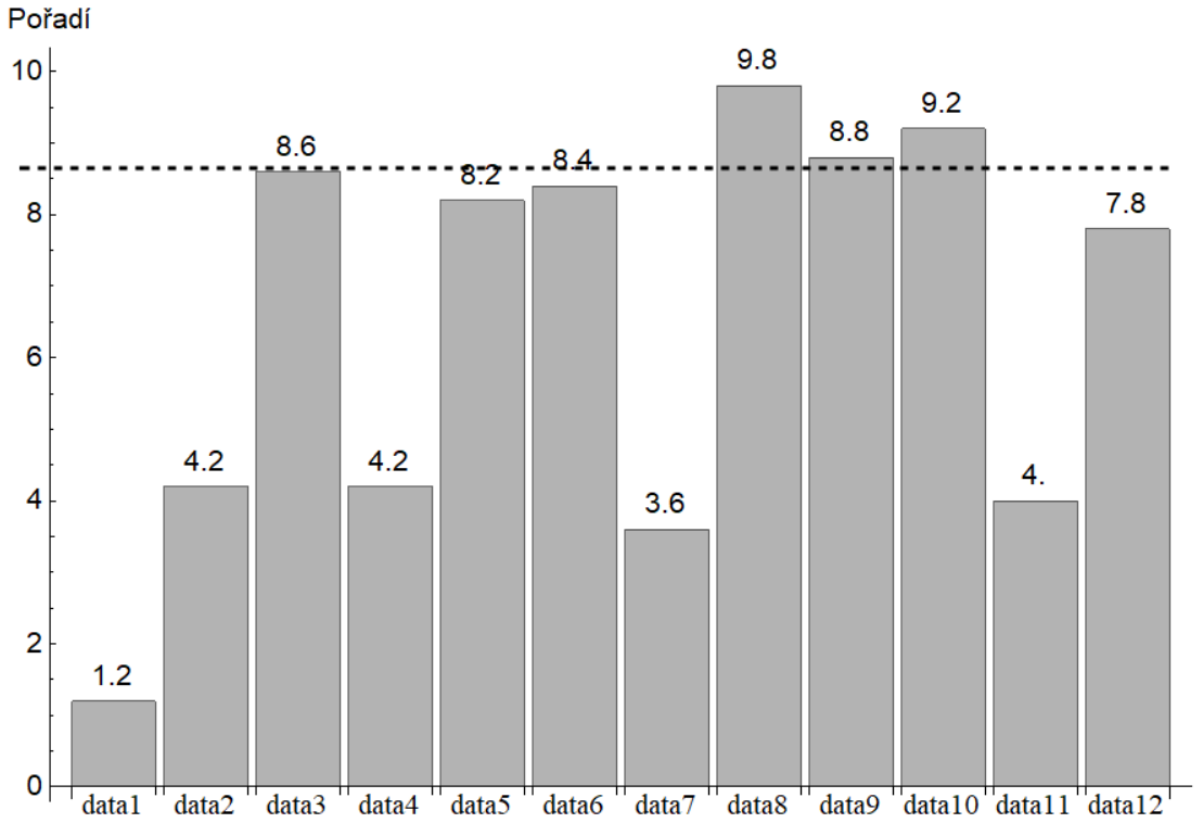
Obr. 99: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_3. [vlastní zdroj]

Na Obr. 99 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku neexistuje varianta algoritmu strojového učení, která by se významně negativně odlišovala od ostatních algoritmů. Oproti tomu, lze identifikovat standardně nastavený LSTM (data2) a LSTM algoritmus nastavený podle RS (data8) jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 114 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_4 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	6.67164	2.03372×10^{-6}

Podle Tab. 114 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



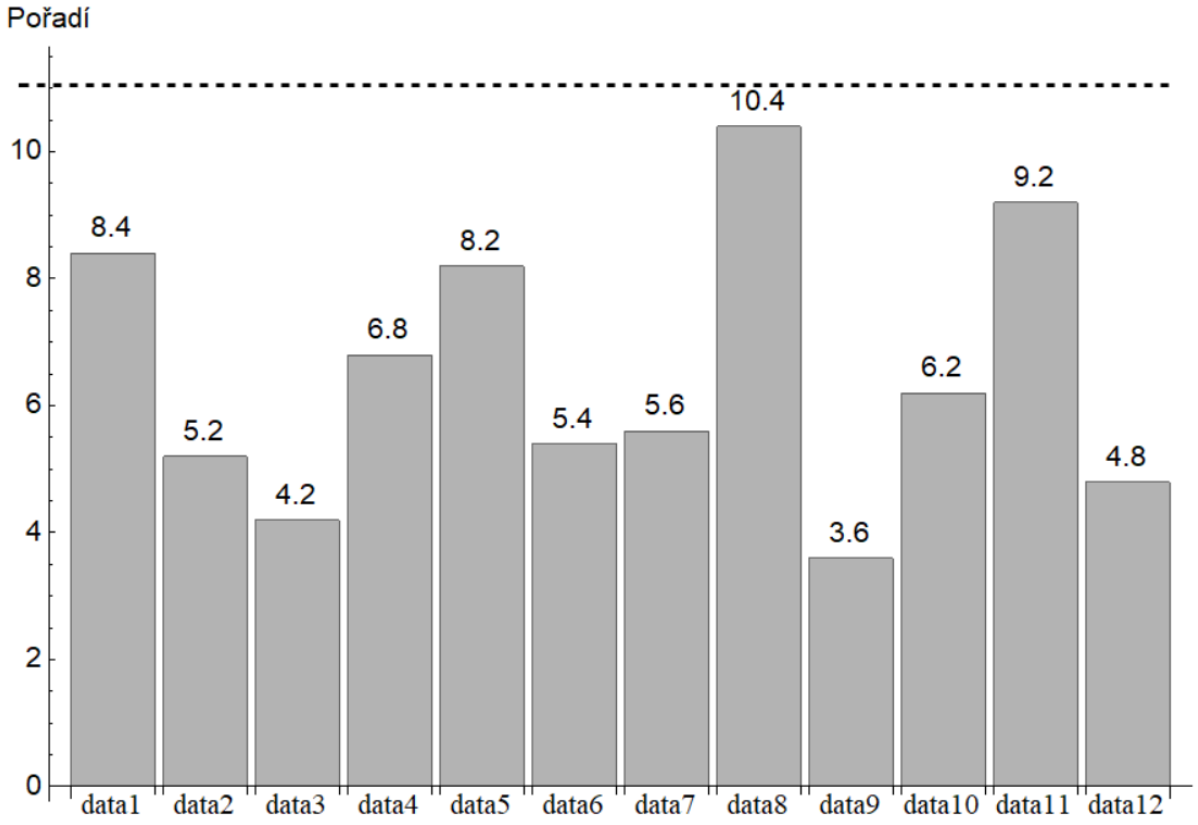
Obr. 100: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_4. [vlastní zdroj]

Na Obr. 100 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmy strojového učení, IF ve standardním nastavení, LSTM a IF nastavené podle RS a neuronová síť nastavená podle TPE mají podstatně horší výsledky oproti ostatním nastavením algoritmů. Oproti tomu lze identifikovat standardně nastavenou neuronovou síť (data1) jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 115 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_5 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.11634	0.0391

Podle Tab. 115 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



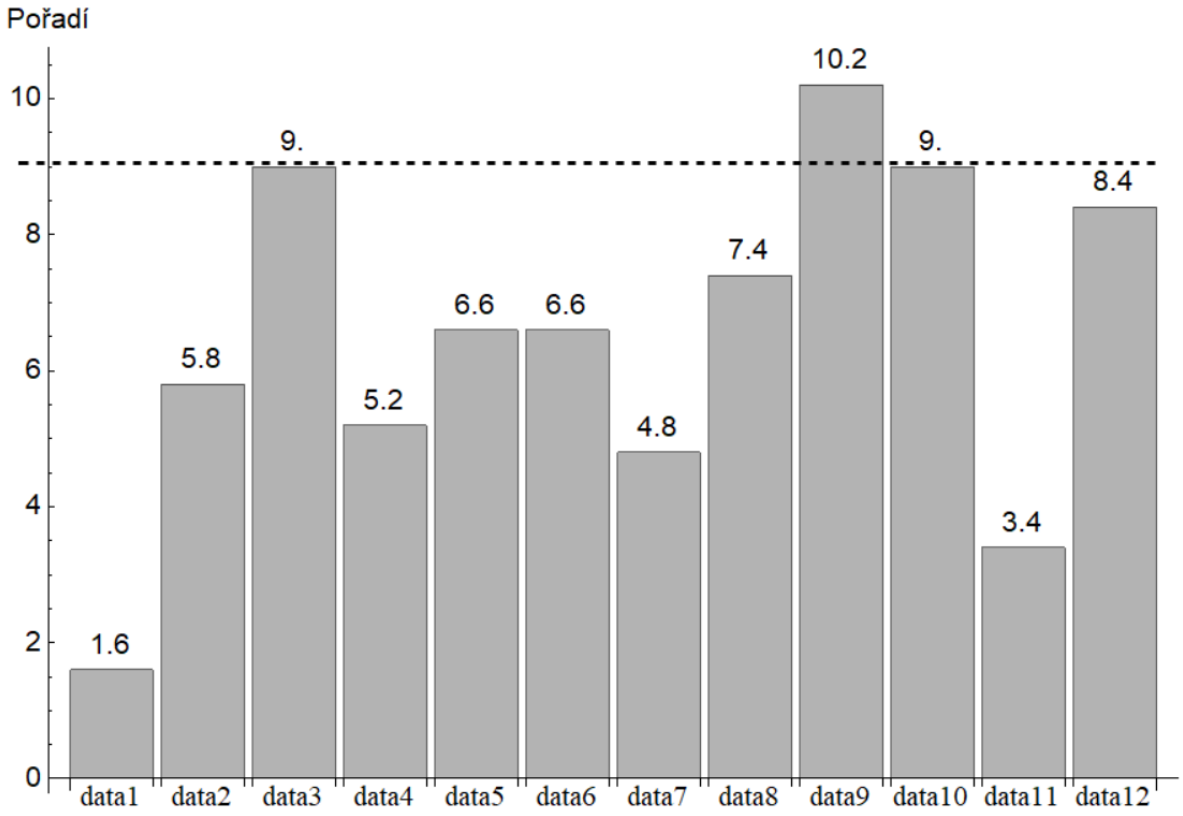
Obr. 101: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_5. [vlastní zdroj]

Na Obr. 101 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku neexistuje varianta (v rámci vybraných algoritmů) algoritmu strojového učení, který by se významně negativně odlišoval od ostatních algoritmů. Oproti tomu lze identifikovat standardně nastavený algoritmus IF, IF nastavený podle RS a IF nastavený podle TPE (data3, data9, data12), jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 116 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_6 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	3.7633	0.00078

Podle Tab. 116 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



Obr. 102: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_6. [vlastní zdroj]

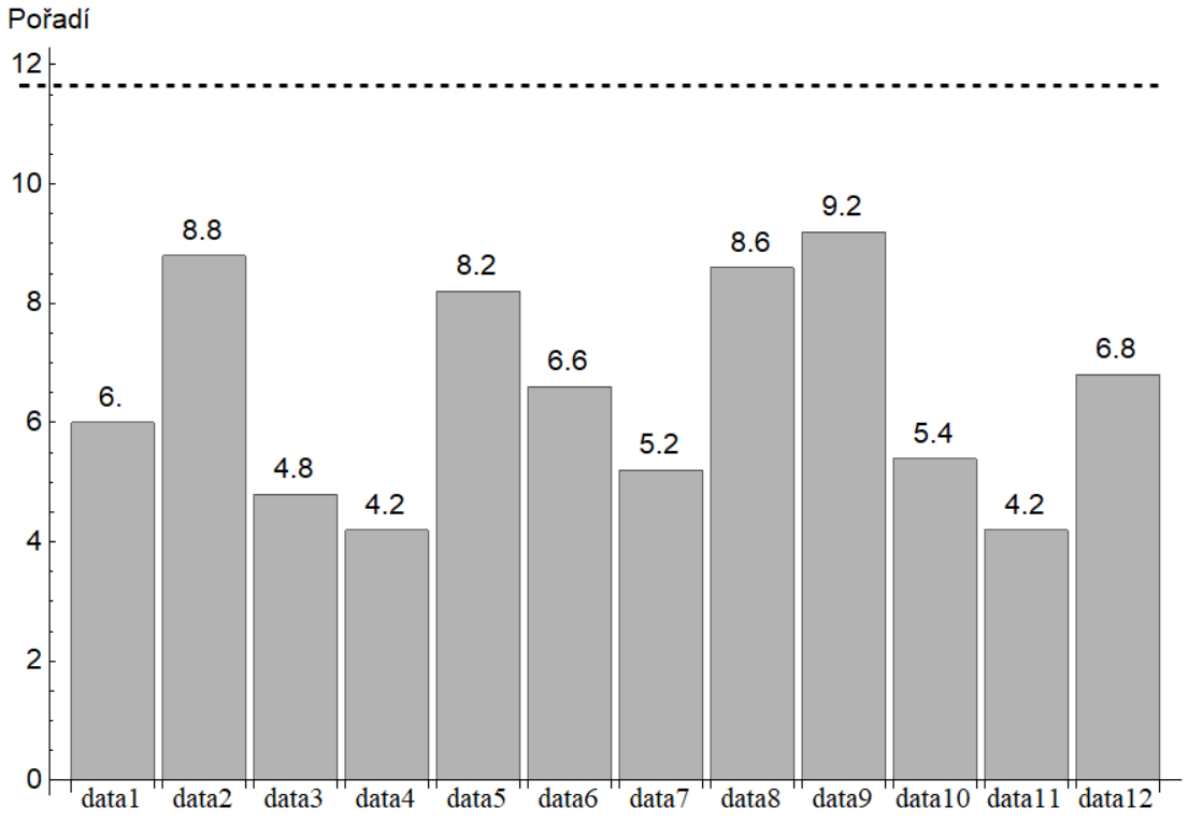
Na Obr. 102 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení, IF nastavený podle RS (data9) má podstatně horší výsledky v rámci tohoto kybernetického útoku ve srovnání s ostatním nastavením algoritmů. Oproti tomu lze identifikovat standardně nastavenou neuronovou síť a algoritmus LSTM nastavený podle TPE, jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Friedmanův test pro jednotlivé algoritmy– dataset 3

Tab. 117 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_1 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.37594	0.2181

Podle Tab. 117 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



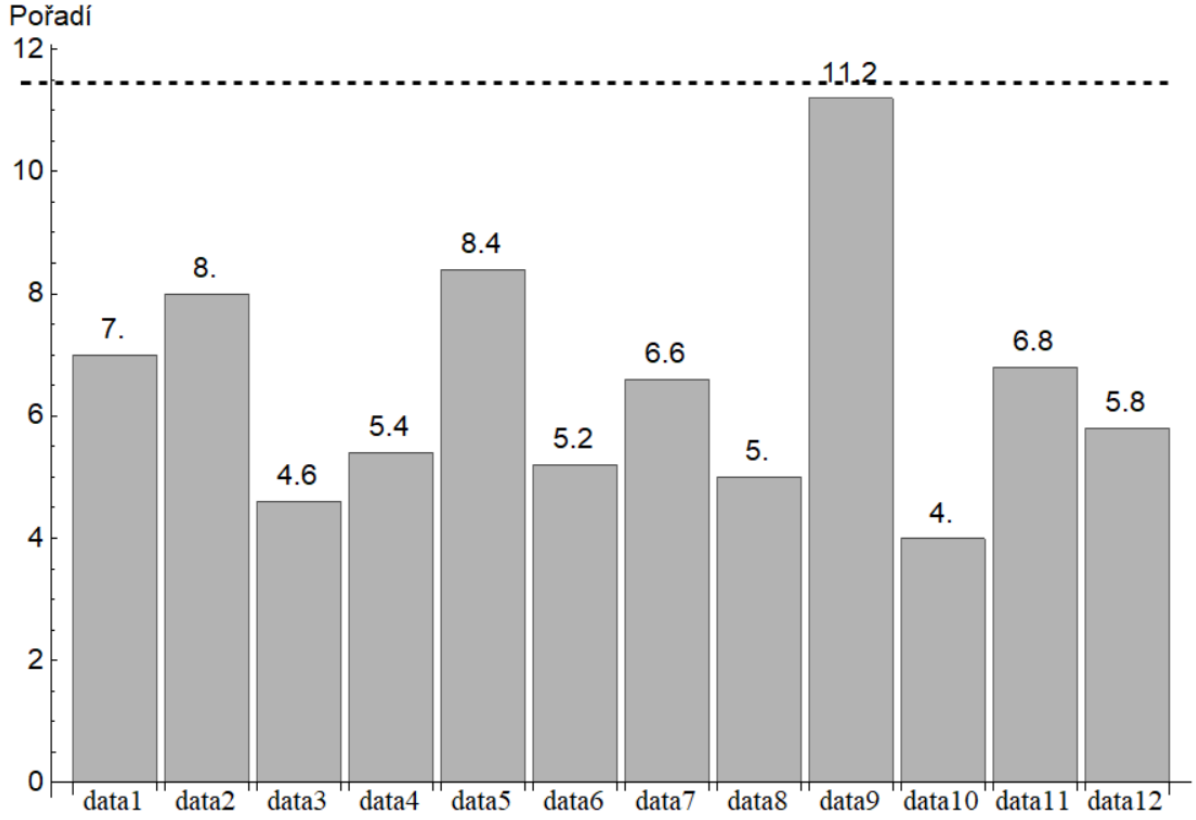
Obr. 103: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_1. [vlastní zdroj]

V rámci Obr. 103 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Tab. 118 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_2 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.76613	0.09007

Podle Tab. 118 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



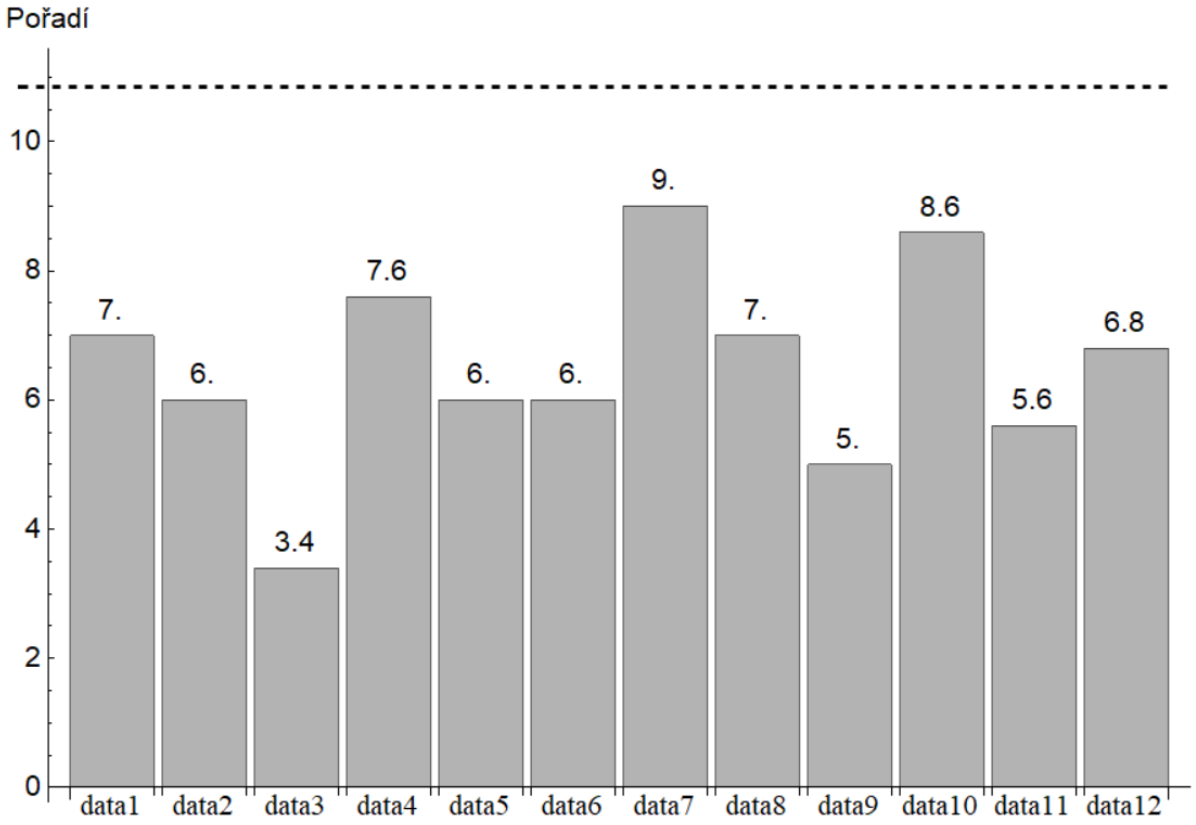
Obr. 104: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_2. [vlastní zdroj]

V rámci Obr. 104 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Tab. 119 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_3 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	0.88388	0.56232

Podle Tab. 119 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



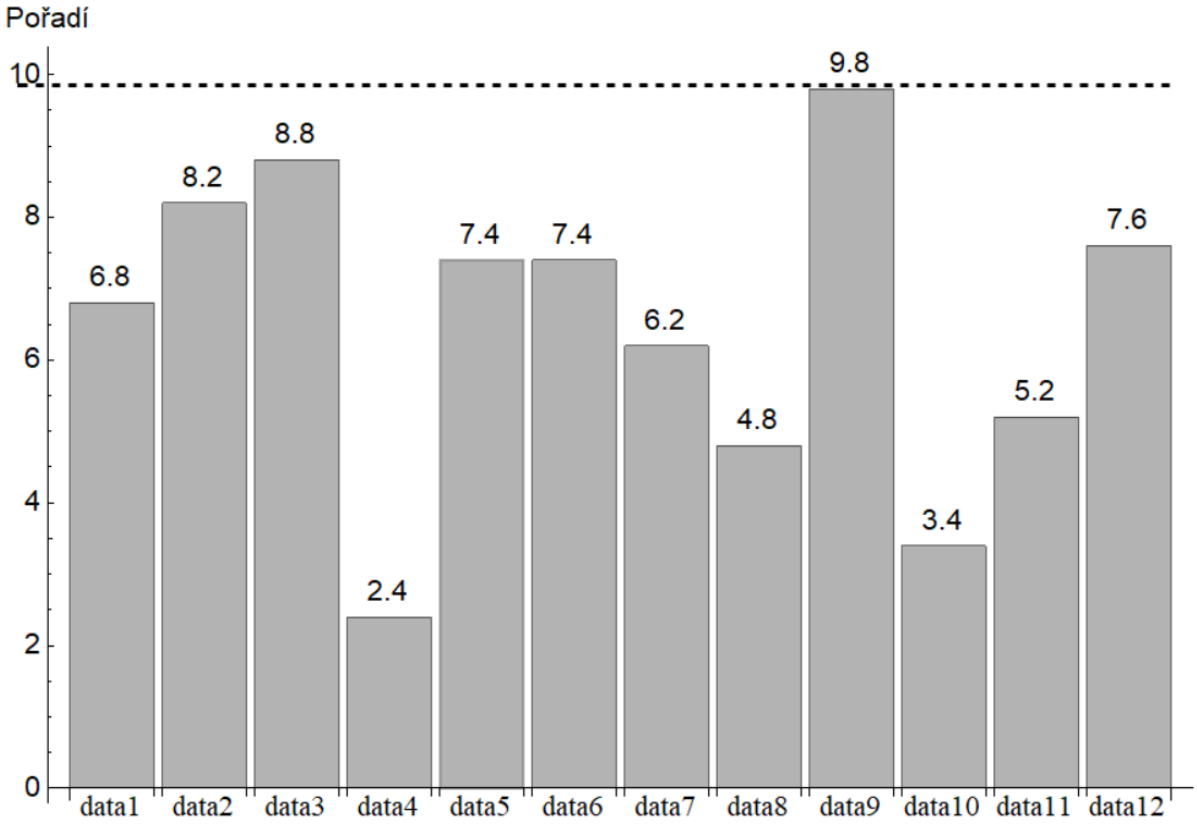
Obr. 105: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_3. [vlastní zdroj]

V rámci Obr. 105 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Tab. 120 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_4 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.36121	0.02164

Podle Tab. 120 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



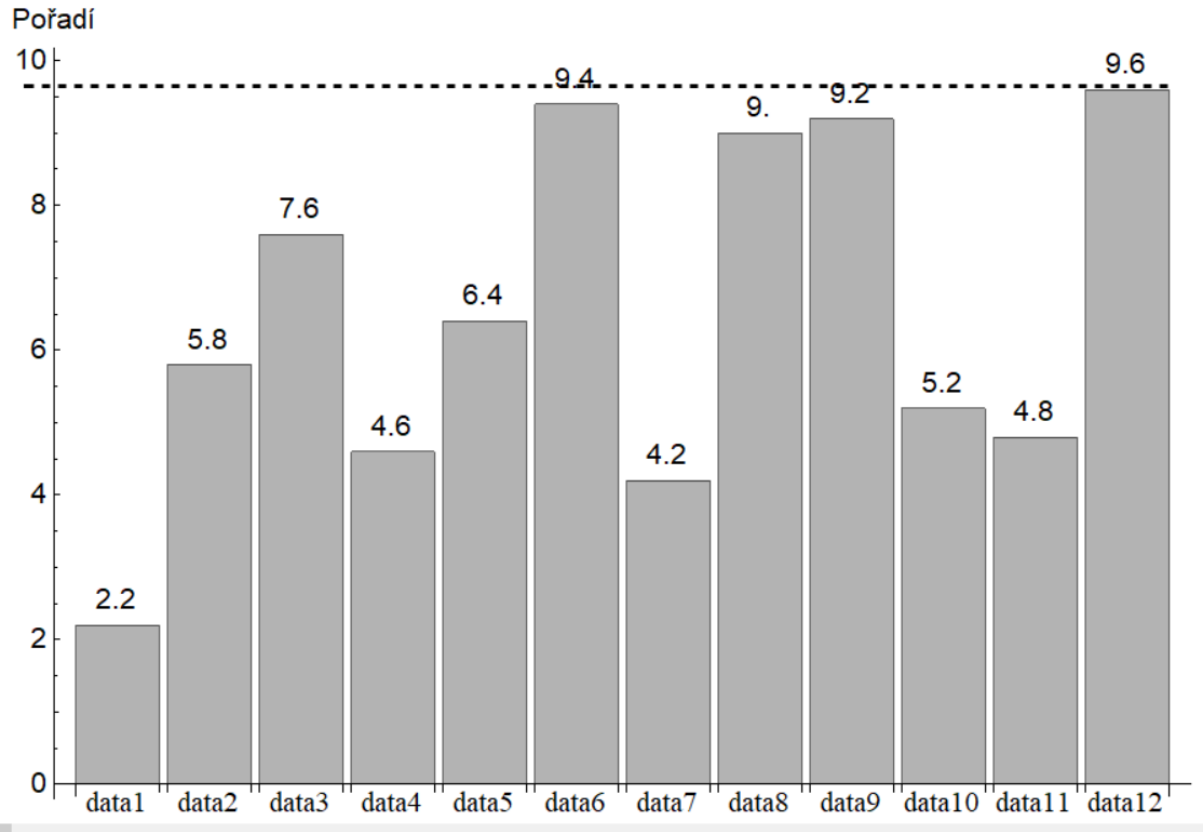
Obr. 106: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_4. [vlastní zdroj]

Na Obr. 106 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení, IF nastavený podle RS (data9) má horší výsledky v rámci tohoto kybernetického útoku vzhledem k ostatním nastavením algoritmů. Oproti tomu, lze identifikovat neuronovou síť nastavenou podle evolučního algoritmu (data4) a neuronovou síť nastavenou podle TPE (dat10), jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 121 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_5 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	3.35597	0.00201

Podle Tab. 121 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



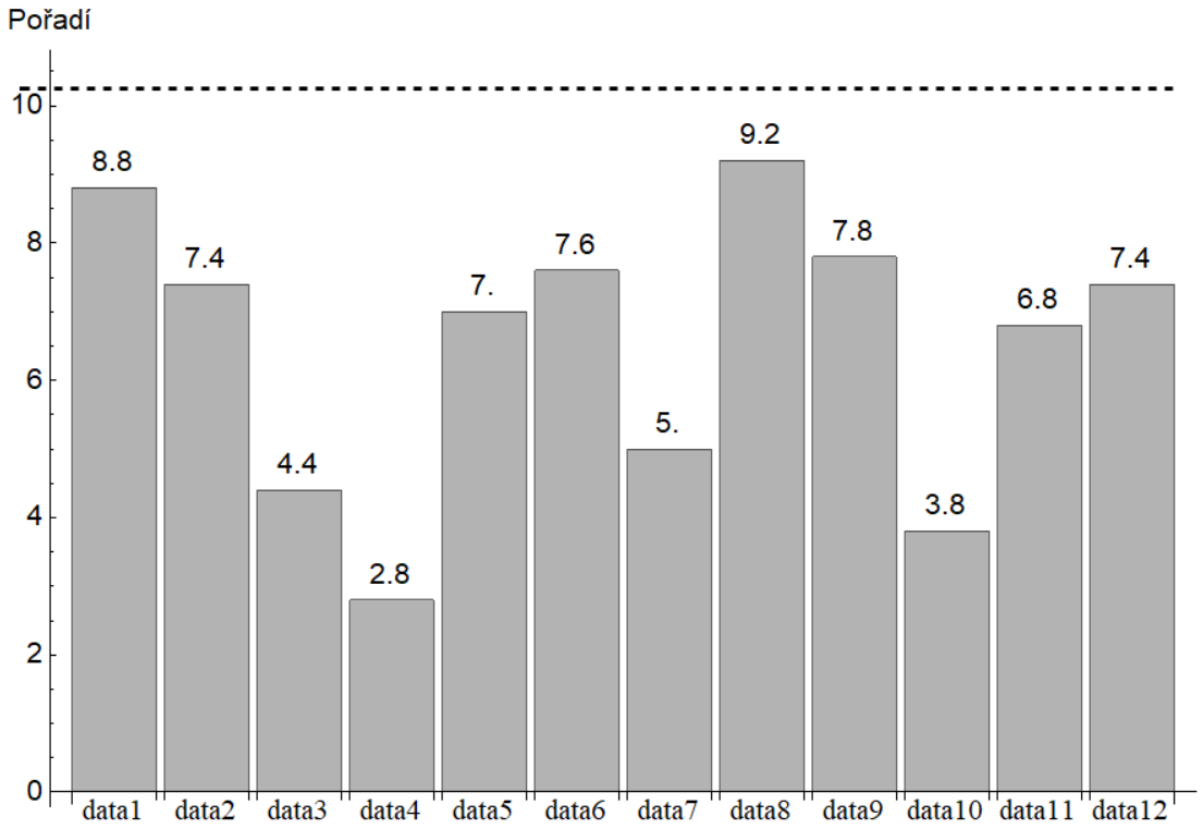
Obr. 107: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_5. [vlastní zdroj]

Na Obr. 107 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku lze konstatovat, že algoritmus strojového učení, IF nastavený podle TPE (data10) má horší výsledky v rámci tohoto kybernetického útoku vzhledem k ostatním nastavením algoritmů. Oproti tomu, lze identifikovat standardně nastavenou neuronovou síť (data1), jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 122 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_6 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.8415	0.07542

Podle Tab. 122 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



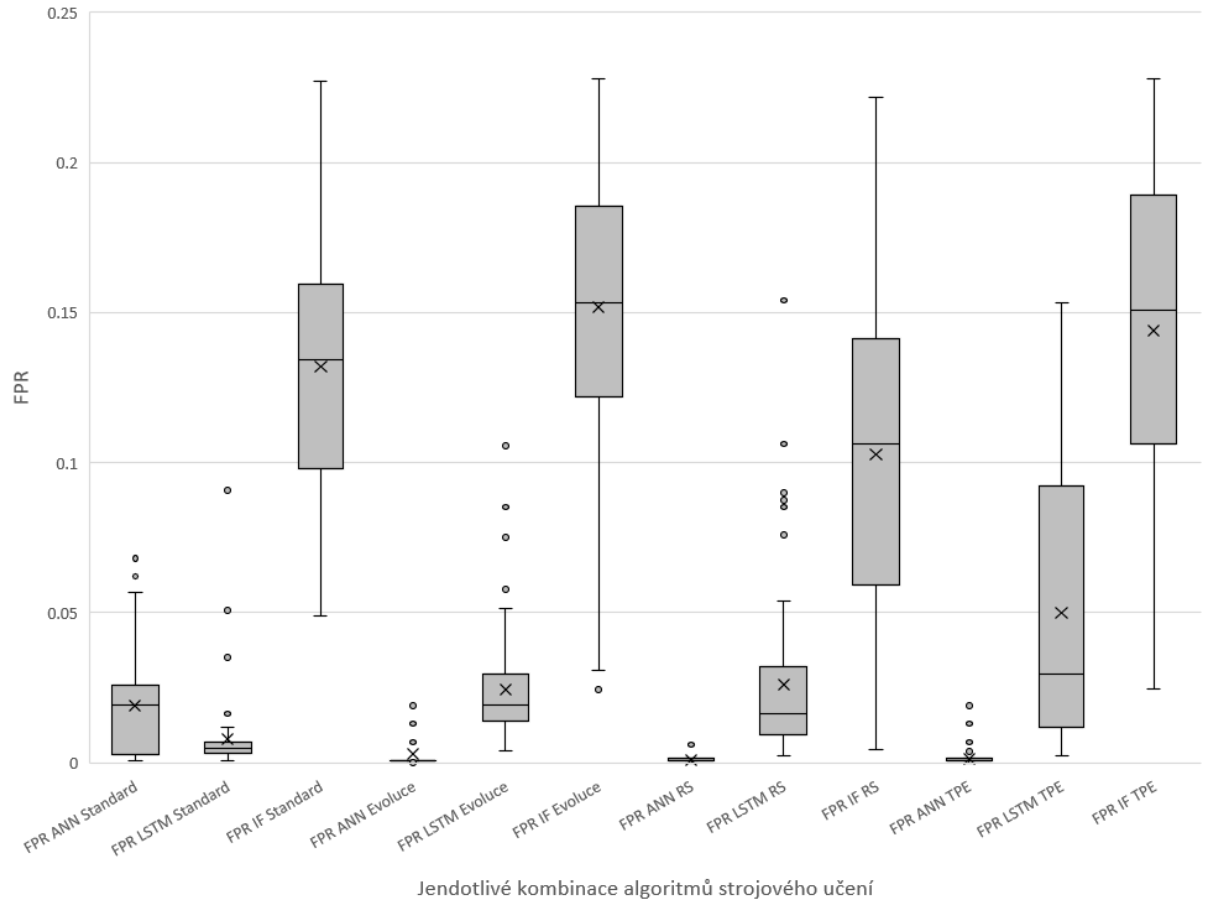
Obr. 108: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_6. [vlastní zdroj]

V rámci Obr. 108 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Příloha F: Porovnání metriky M_{FPR} pro jednotlivá řešení.

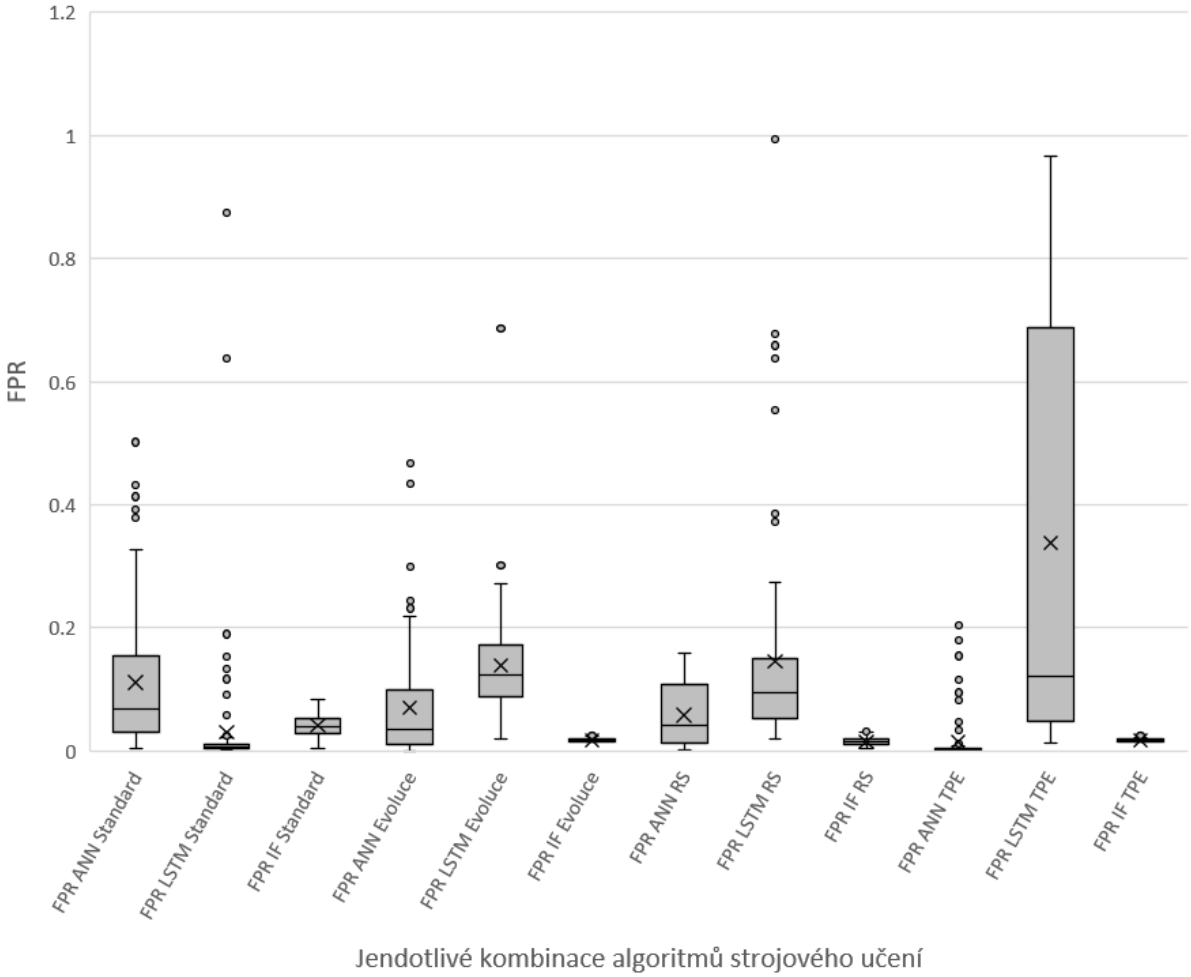
Porovnání metriky M_{FPR} pro jednotlivá řešení – dataset 1

Na Obr. 109 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA1_1. Z výsledků lze vyvodit tři nejlepší zástupce, kteří se odlišují od ostatních zástupců. Tyto algoritmy mají v podstatě totožné výsledky. Jedná se o neuronovou síť nastavenou podle evolučního algoritmu (medián: 0.00077), neuronovou síť nastavenou pomocí optimalizačního algoritmu RS (medián: 0.00077) a neuronovou síť, která je nastavena pomocí TPE (medián: 0.00077).



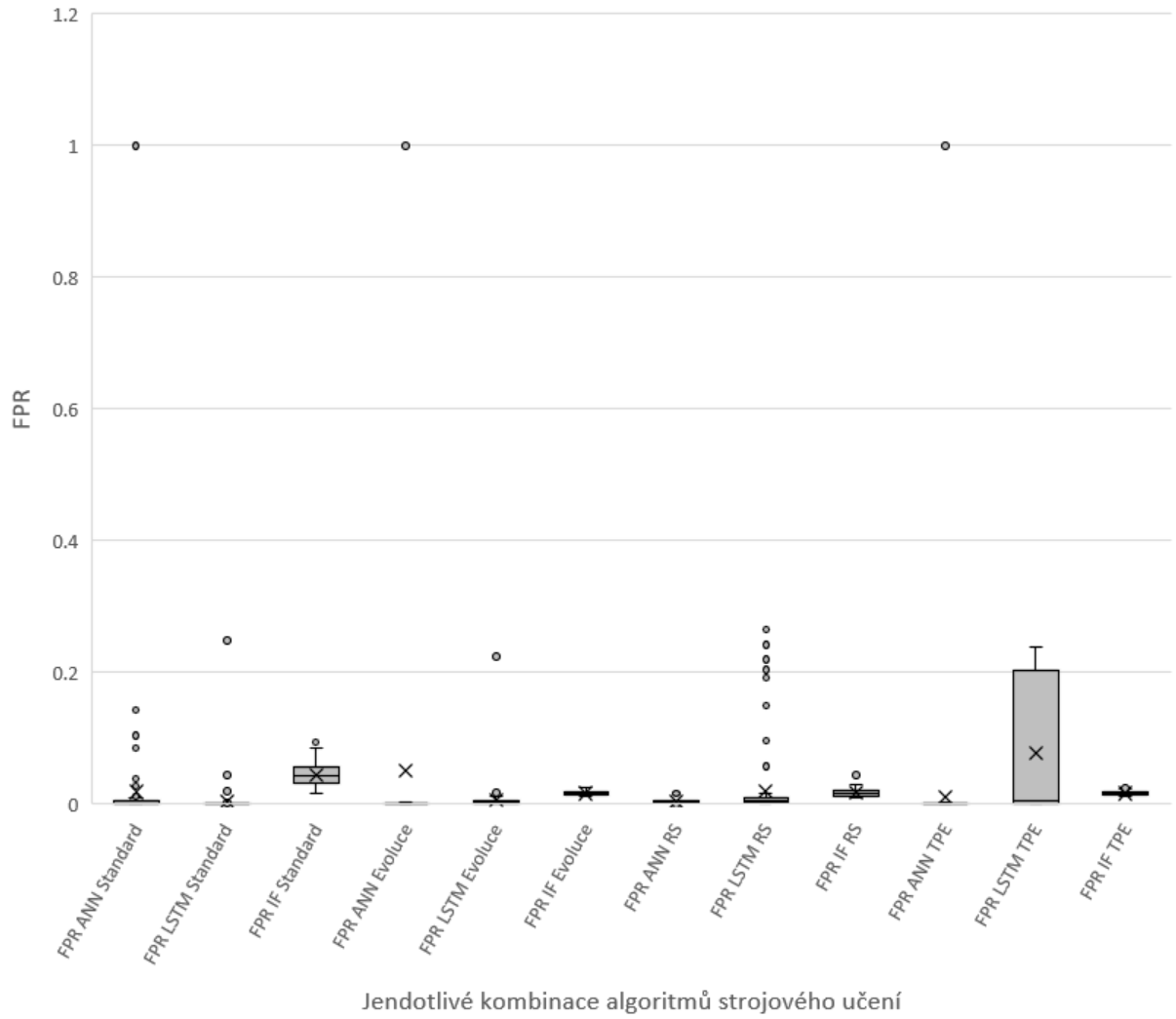
Obr. 109: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA1_1. [vlastní zdroj]

Na Obr. 110 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA1_2. Z výsledků lze identifikovat dva nejlepší zástupce, kteří se odlišují od ostatních zástupců. Nejlepším zástupcem je neuronová síť nastavena pomocí optimalizačního algoritmu TPE (medián: 0.00152). Druhým nejlepším zástupcem je standardně nastavený LSTM (medián: 0.00666).



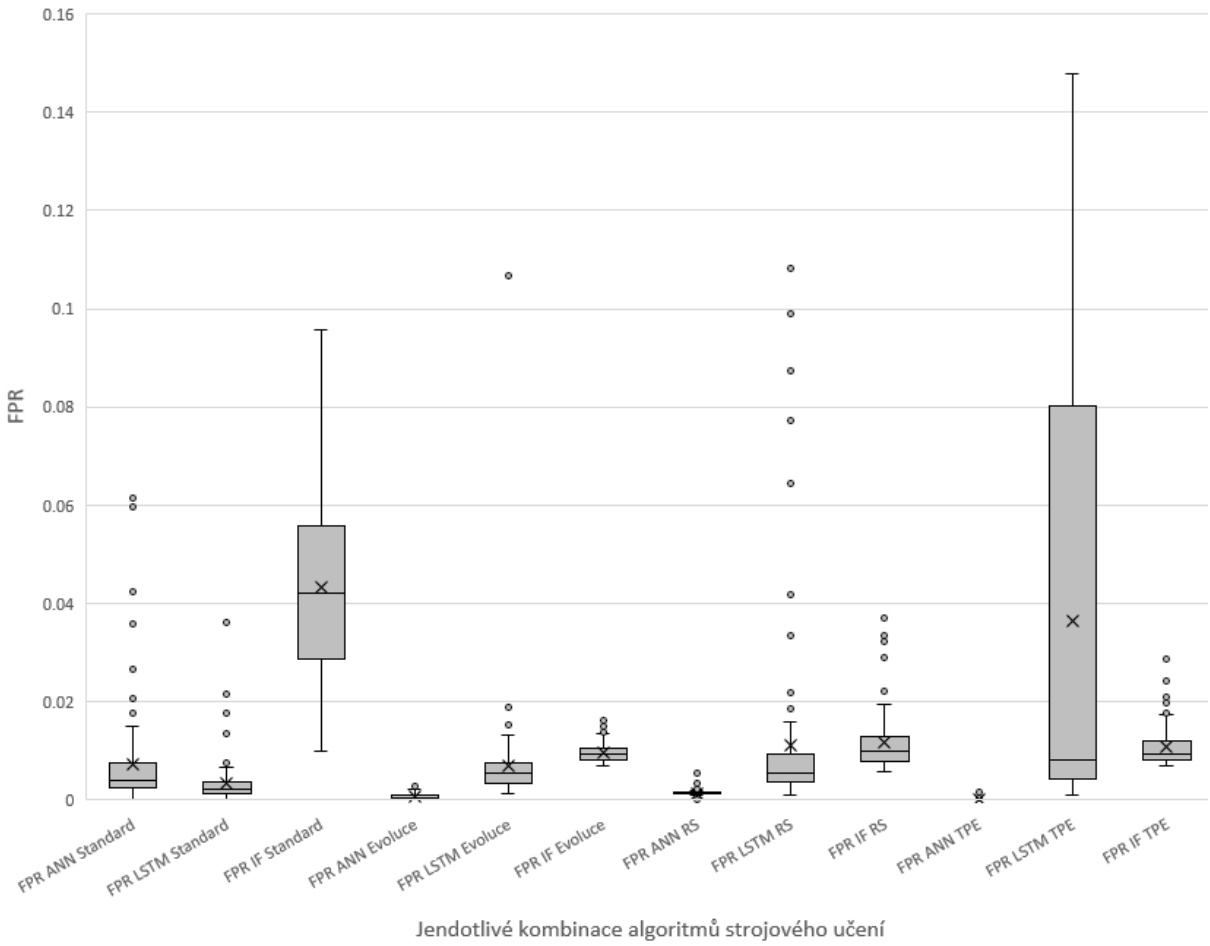
Obr. 110: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA1_2. [vlastní zdroj]

Na Obr. 111 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA1_3. Z výsledků lze identifikovat dva nejlepší zástupce, kteří se odlišují od ostatních zástupců. Nejlepším zástupcem je neuronová síť nastavená pomocí optimalizačního algoritmu TPE (medián: 0.00027). Druhým nejlepším zástupcem je standardně nastavený LSTM (medián: 0.00045).



Obr. 111: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA1_3. [vlastní zdroj]

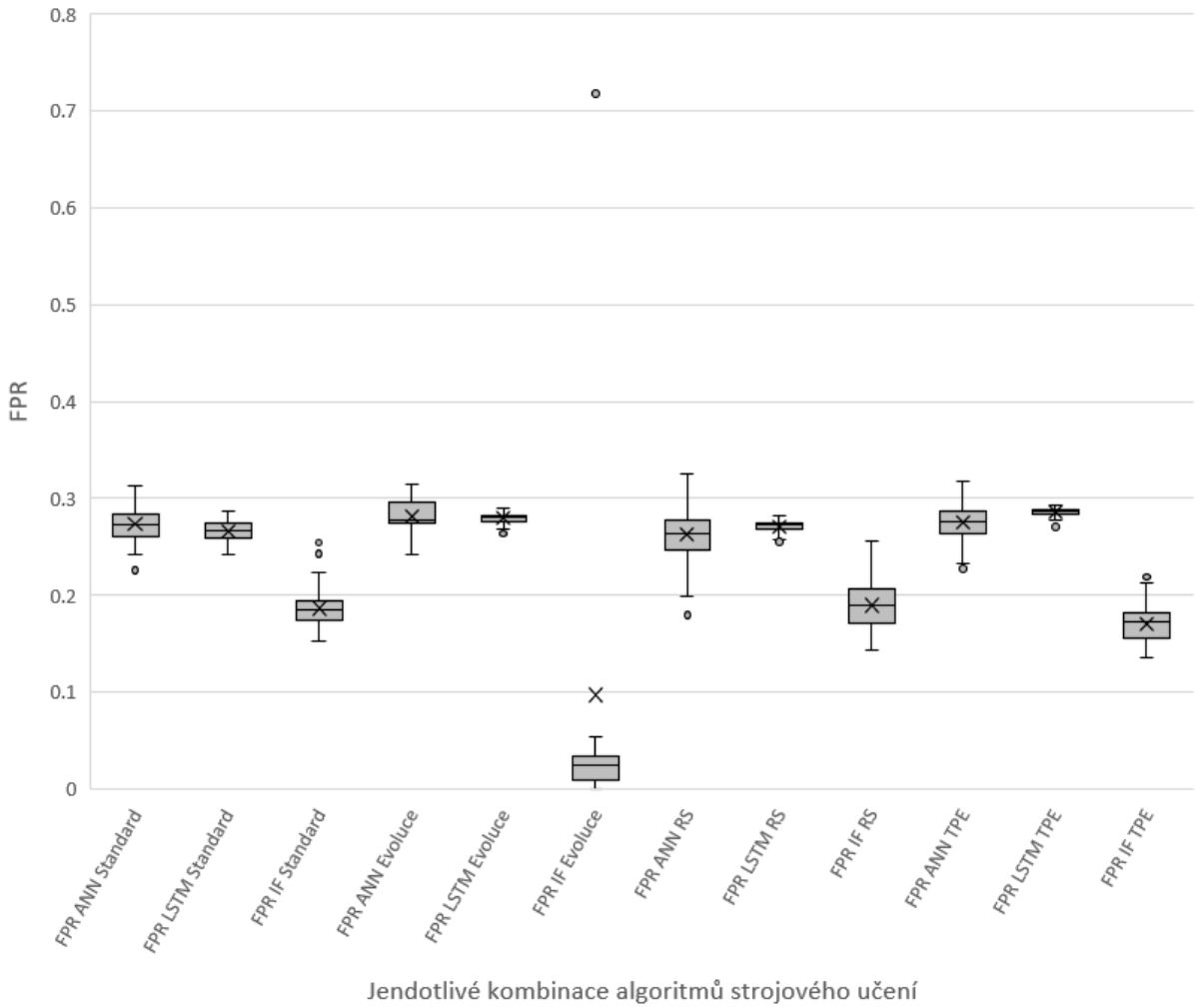
Na Obr. 112 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA1_4. Z výsledků lze vyvodit dva nejlepší zástupce, kteří se odlišují od ostatních zástupců. Oba tito zástupci využívají neuronovou síť. V prvním případě je tento algoritmus strojového učení nastaven pomocí optimalizačního algoritmu TPE (medián: 0). Ve druhém případě byl algoritmus nastaven pomocí evolučního algoritmu (medián: 0.00031).



Obr. 112: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA1_4. [vlastní zdroj]

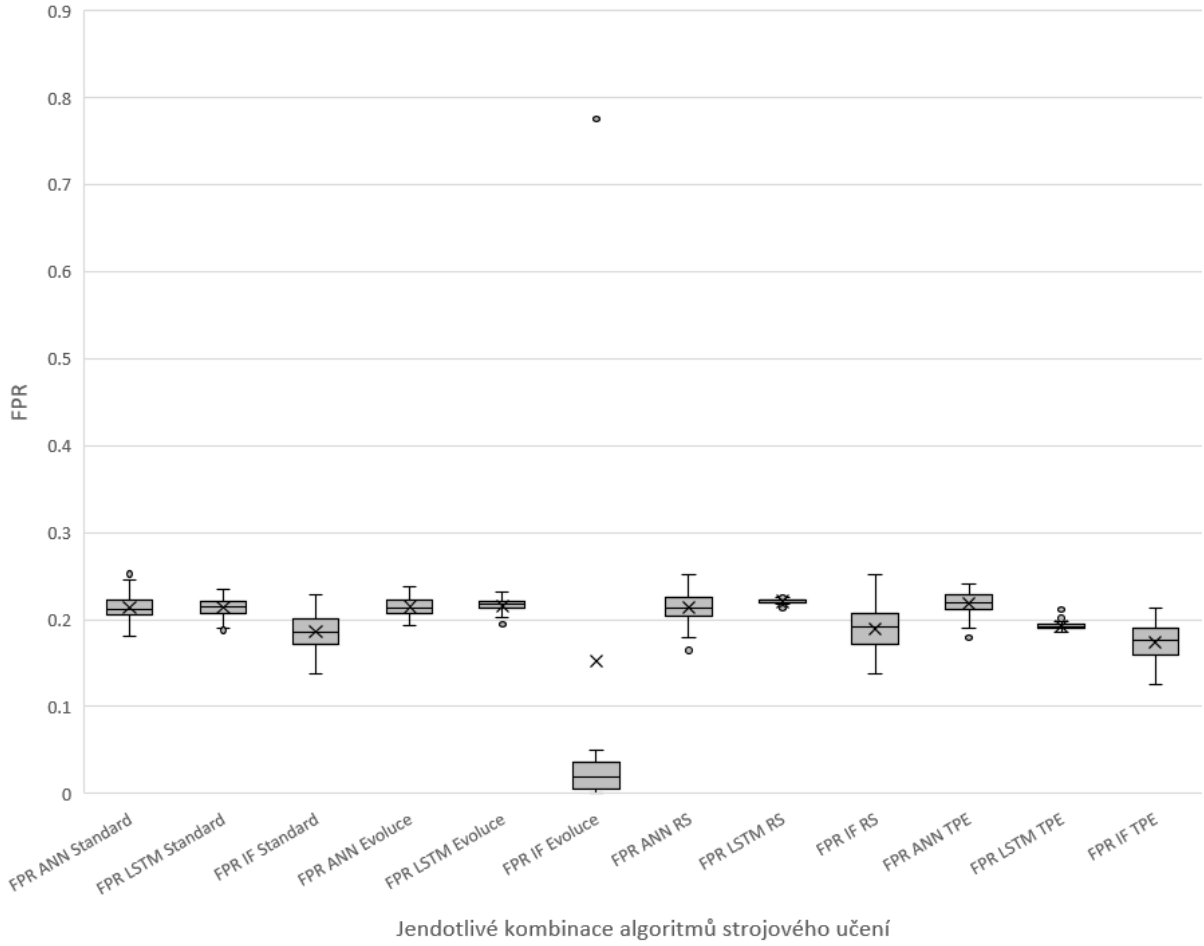
Porovnání metriky M_{FPR} pro jednotlivá řešení – dataset 2

Na Obr. 113 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_1. Z výsledků lze vybrat nejlepšího zástupce, který se výrazně odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.02444).



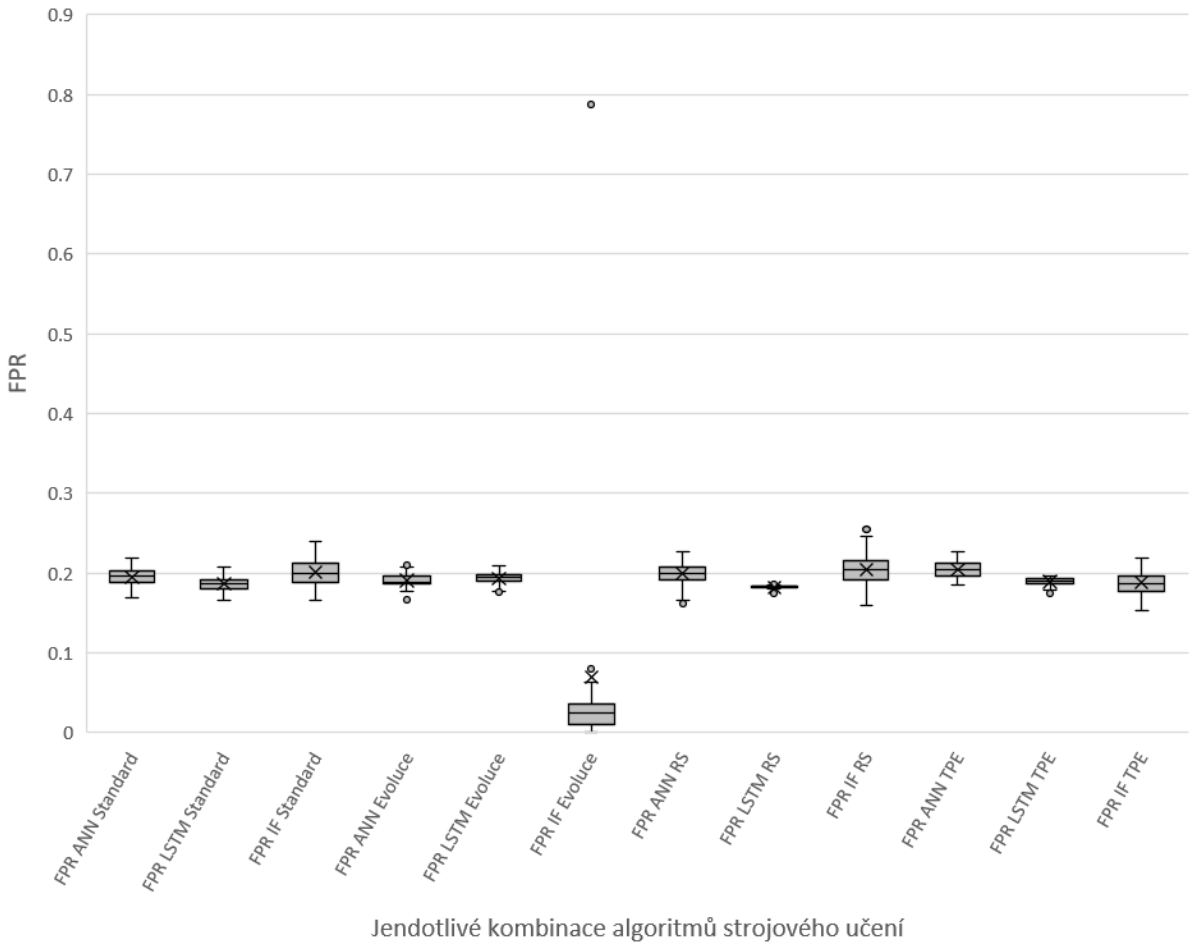
Obr. 113: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_1. [vlastní zdroj]

Na Obr. 114 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_2. Z výsledků lze vybrat nejlepšího zástupce, který se výrazně odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.01972).



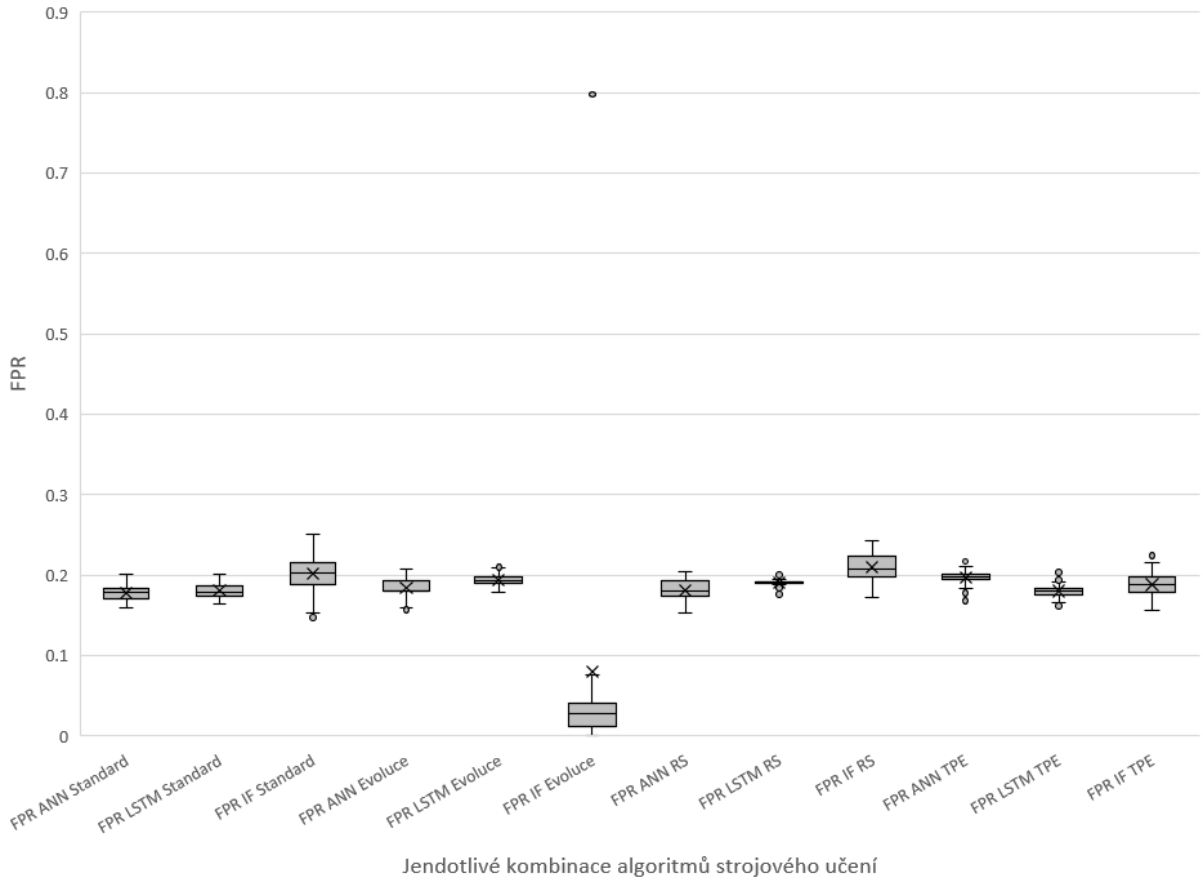
Obr. 114: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_2. [vlastní zdroj]

Na Obr. 115 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_3. Z výsledků lze vybrat nejlepšího zástupce, který se výrazně odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.02557).



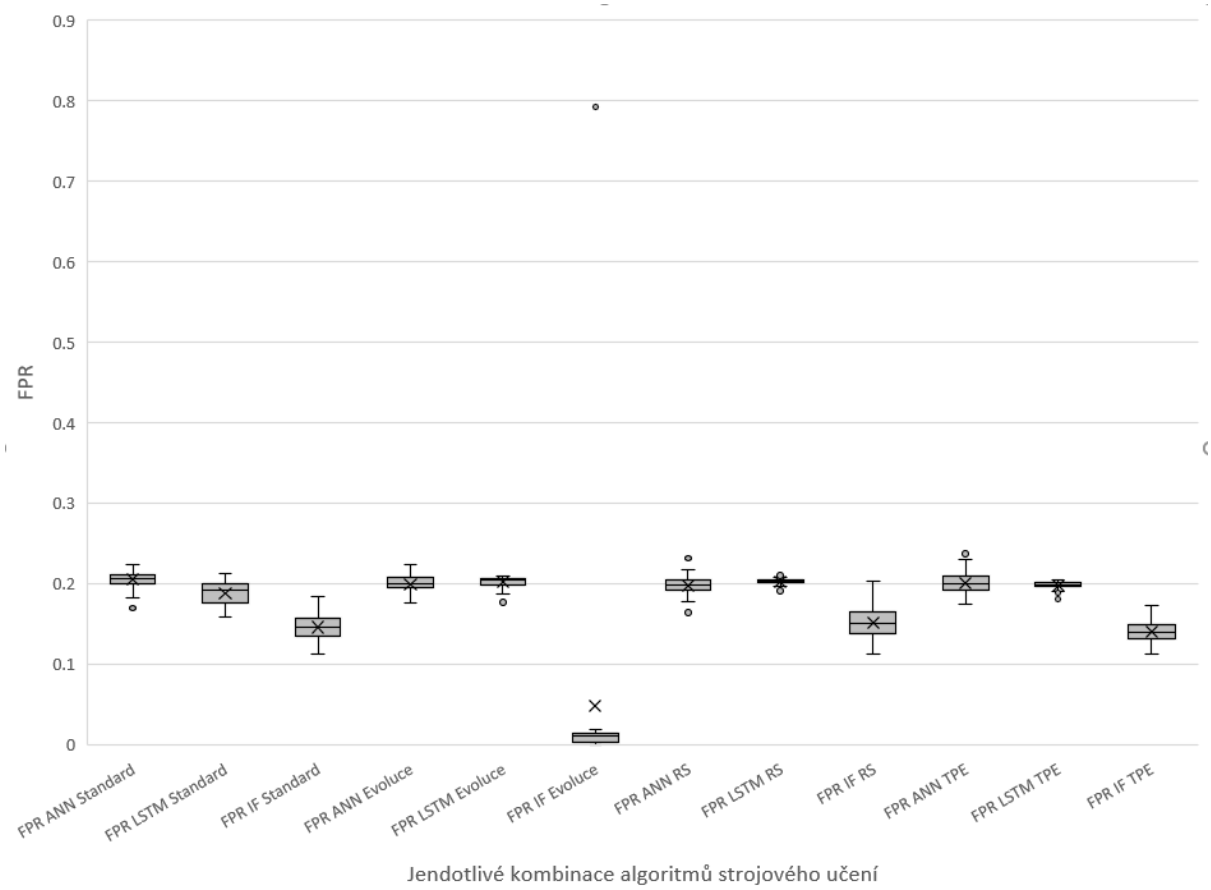
Obr. 115: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_3. [vlastní zdroj]

Na Obr. 116 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_4. Z výsledků lze vybrat nejlepšího zástupce, který se výrazně odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.02726).



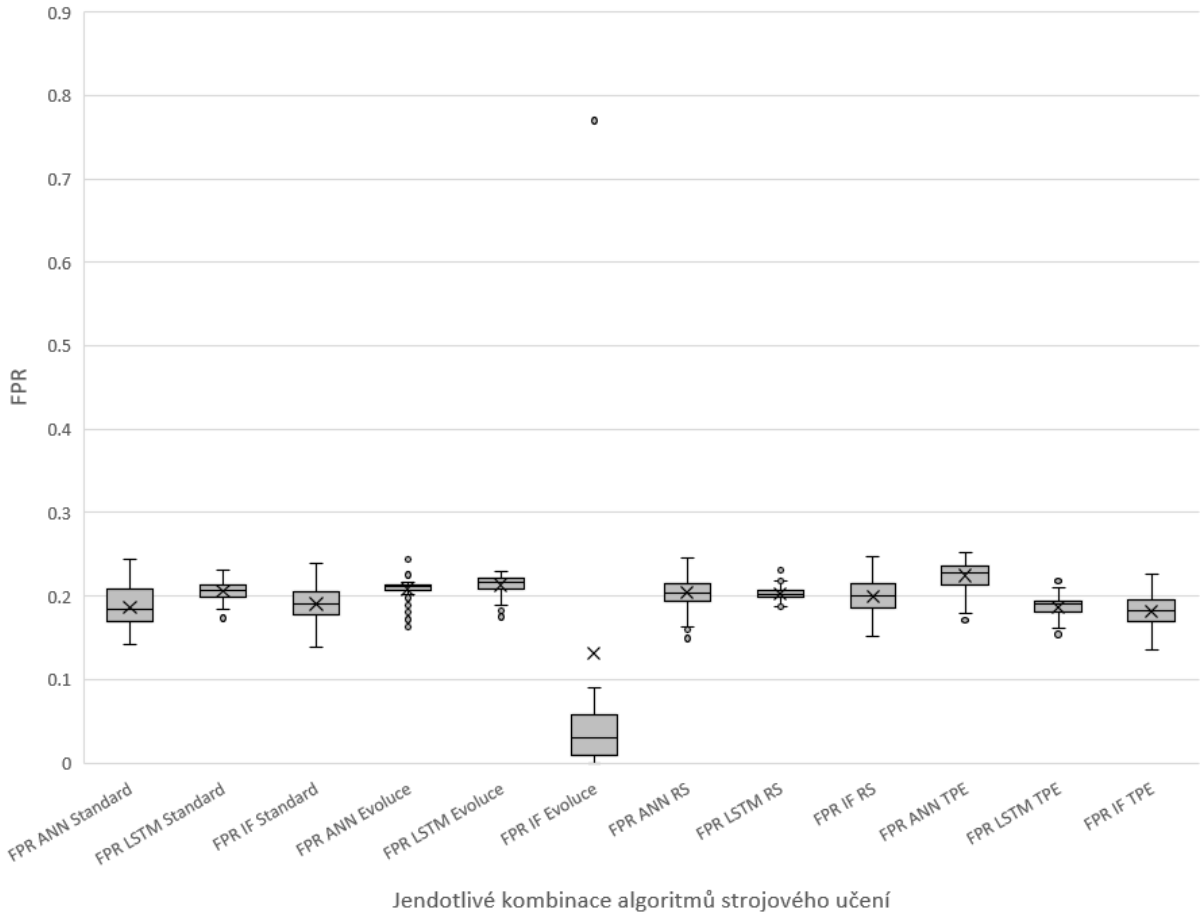
Obr. 116: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_4. [vlastní zdroj]

Na Obr. 117 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_5. Z výsledků lze vybrat nejlepšího zástupce, který se výrazně odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.01124).



Obr. 117: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_5. [vlastní zdroj]

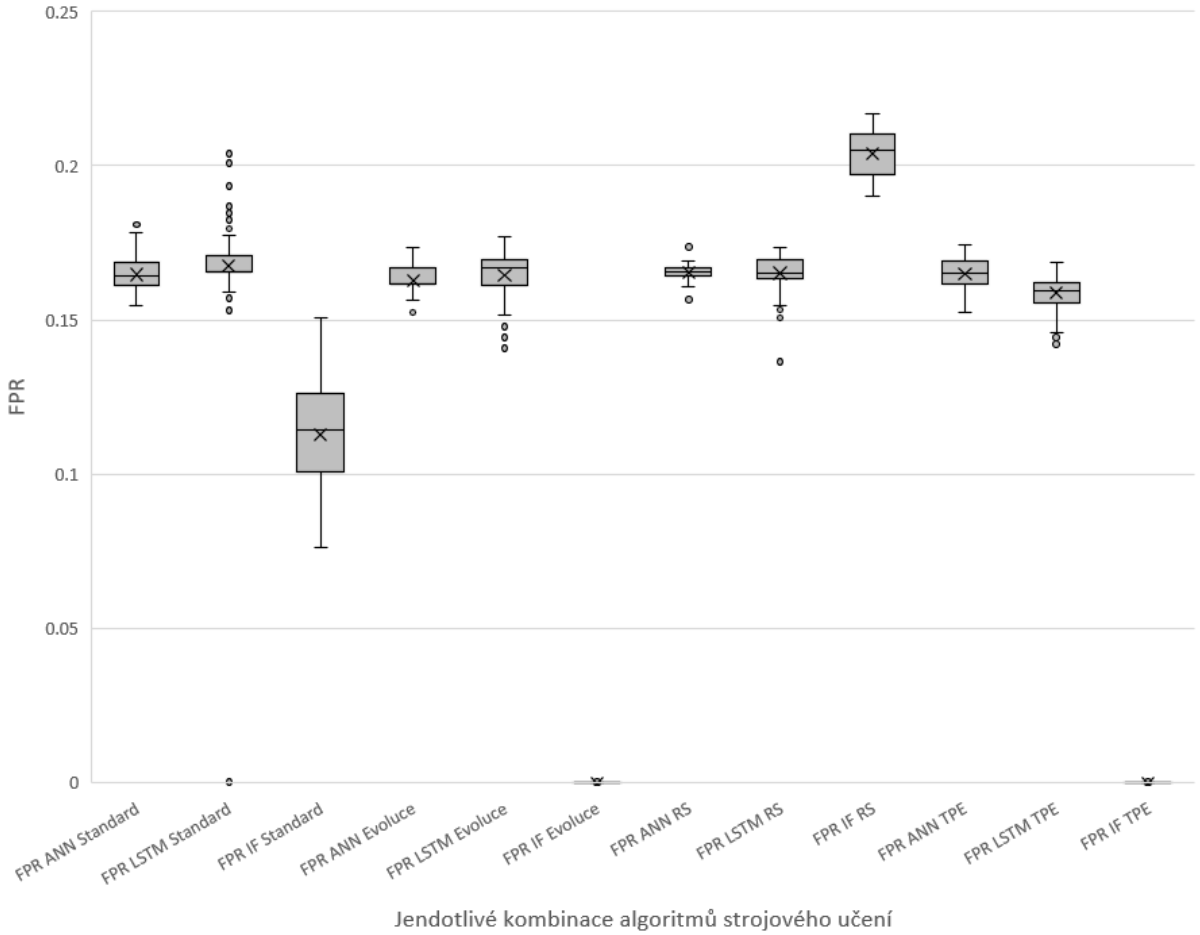
Na Obr. 118 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_6. Z výsledků lze vybrat nejlepšího zástupce, který se výrazně odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.03095).



Obr. 118: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_6. [vlastní zdroj]

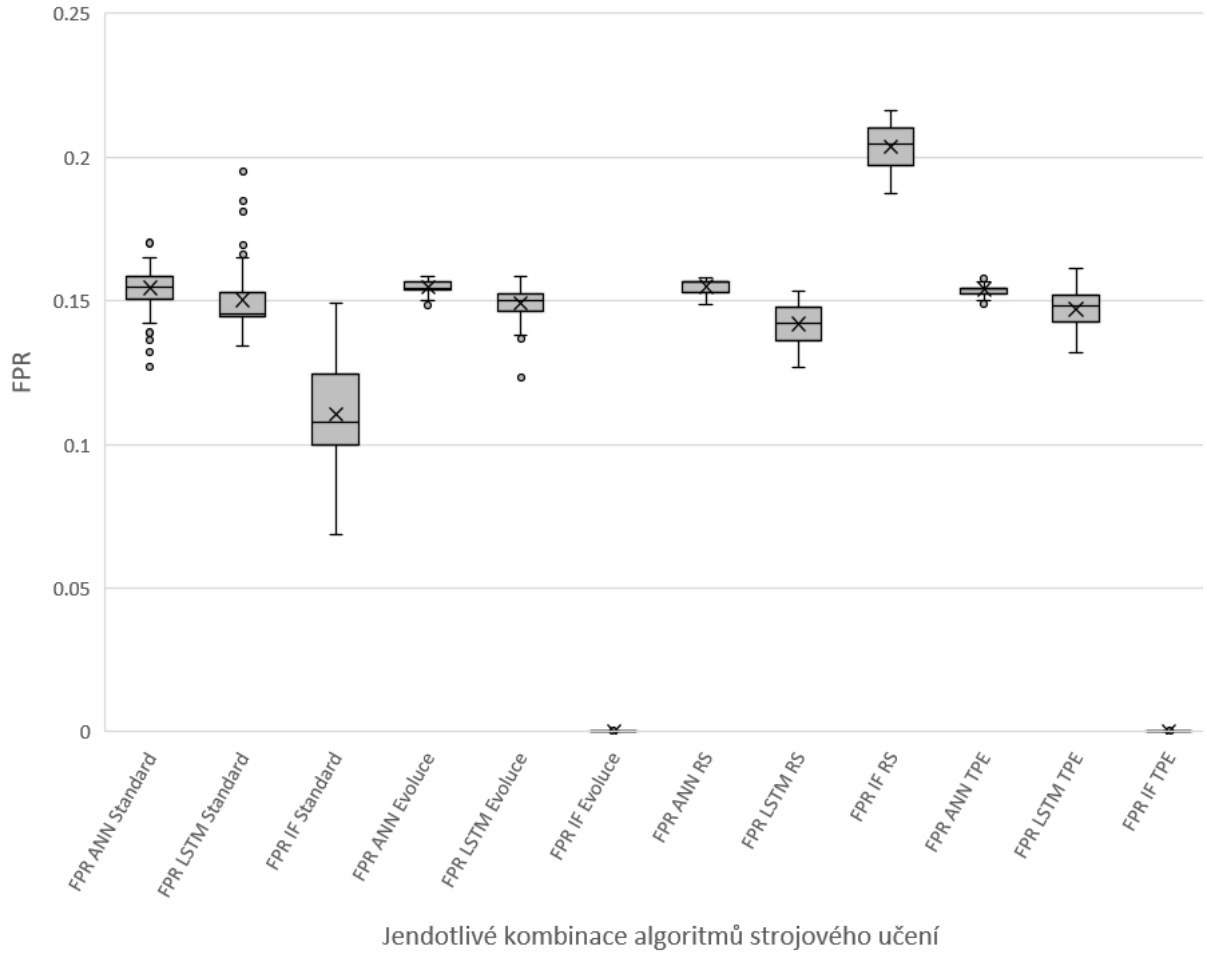
Porovnání metriky M_{FPR} pro jednotlivá řešení – dataset 3

Na Obr. 119 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_1. Z výsledků lze vybrat dva nejlepší zástupce, kteří se výrazně odlišují od ostatních zástupců. Oba zástupci (IF nastavený podle evolučního algoritmu a IF, který je nastaven podle TPE) dosahují v podstatě stejného výsledku, kde prakticky metrika M_{FPR} dosahuje nulové hodnoty (medián = 0).



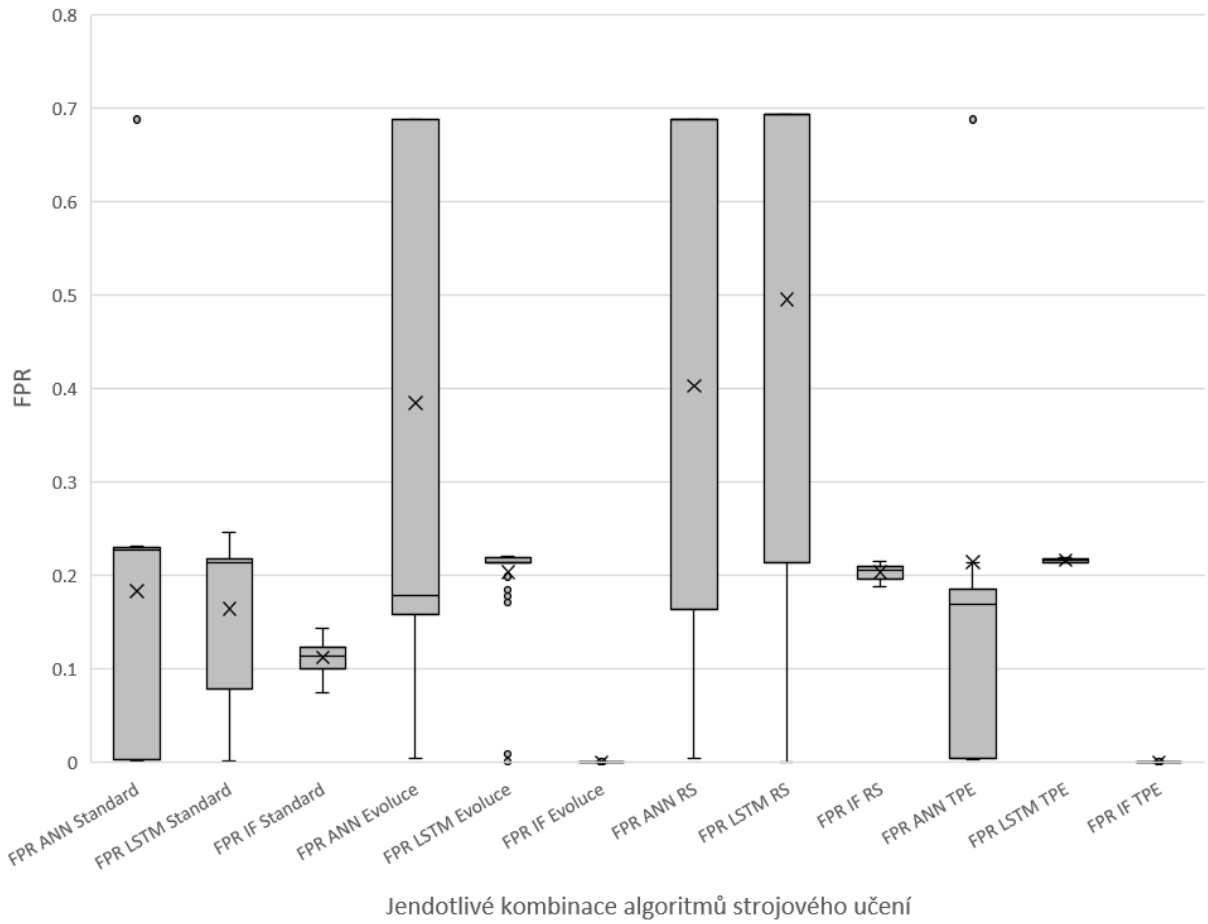
Obr. 119: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_1. [vlastní zdroj]

Na Obr. 120 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_2. Z výsledků lze vybrat dva nejlepší zástupce, kteří se výrazně odlišují od ostatních zástupců. Oba zástupci (IF nastavený podle evolučního algoritmu a IF, který je nastaven podle TPE) dosahují v podstatě stejného výsledku, kde prakticky metrika M_{FPR} dosahuje nulové hodnoty (medián = 0).



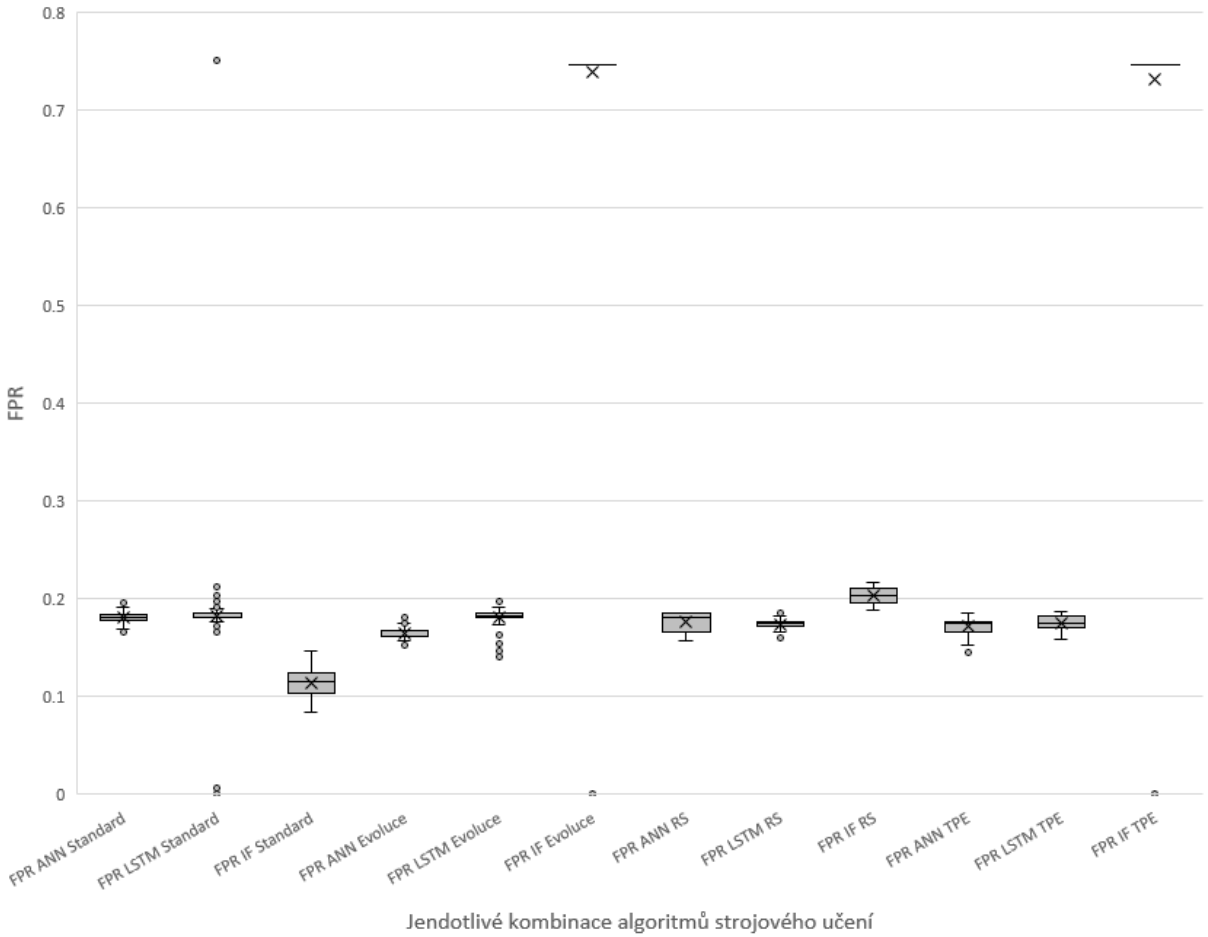
Obr. 120: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_2. [vlastní zdroj]

Na Obr. 121 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_3. Z výsledků lze vybrat dva nejlepší zástupce, kteří se výrazně odlišují od ostatních zástupců. Oba zástupci (IF nastavený podle evolučního algoritmu a IF, který je nastaven podle TPE) dosahují v podstatě stejného výsledku, kde prakticky metrika M_{FPR} dosahuje nulové hodnoty (medián = 0).



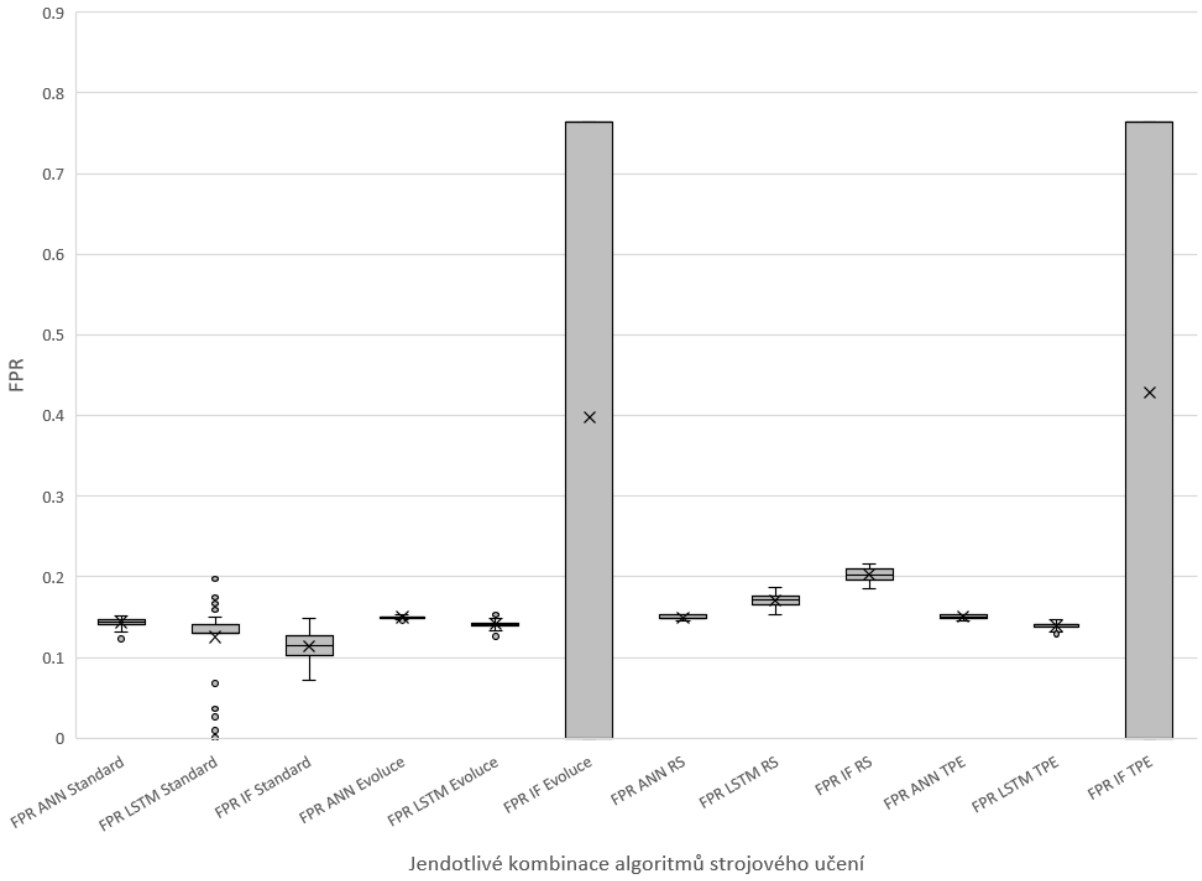
Obr. 121: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_3. [vlastní zdroj]

Na Obr. 122 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_4. Výsledky byly v tomto případě nepřesné (chybné). V rámci tohoto kybernetického útoku oba algoritmy (IF nastavený podle evolučního algoritmu a IF, který je nastaven podle TPE) ve skutečnosti měli M_{FPR} nulový. Avšak v tomto případě také nenašly žádný bod tohoto kybernetické útoku. Z tohoto důvodu má pozitivní třída (True positive) nulovou hodnotu. Tato hodnota představuje však problém z pohledu kalkulace dílčích metrik jako je M_{FPR}, jelikož se vyskytuje ve výpočtu těchto metrik. Z tohoto důvodu výsledky ve zmíněných případech nabývají poměrně velkých hodnot, přitom jejich reálná hodnota metriky M_{FPR} je nulová.



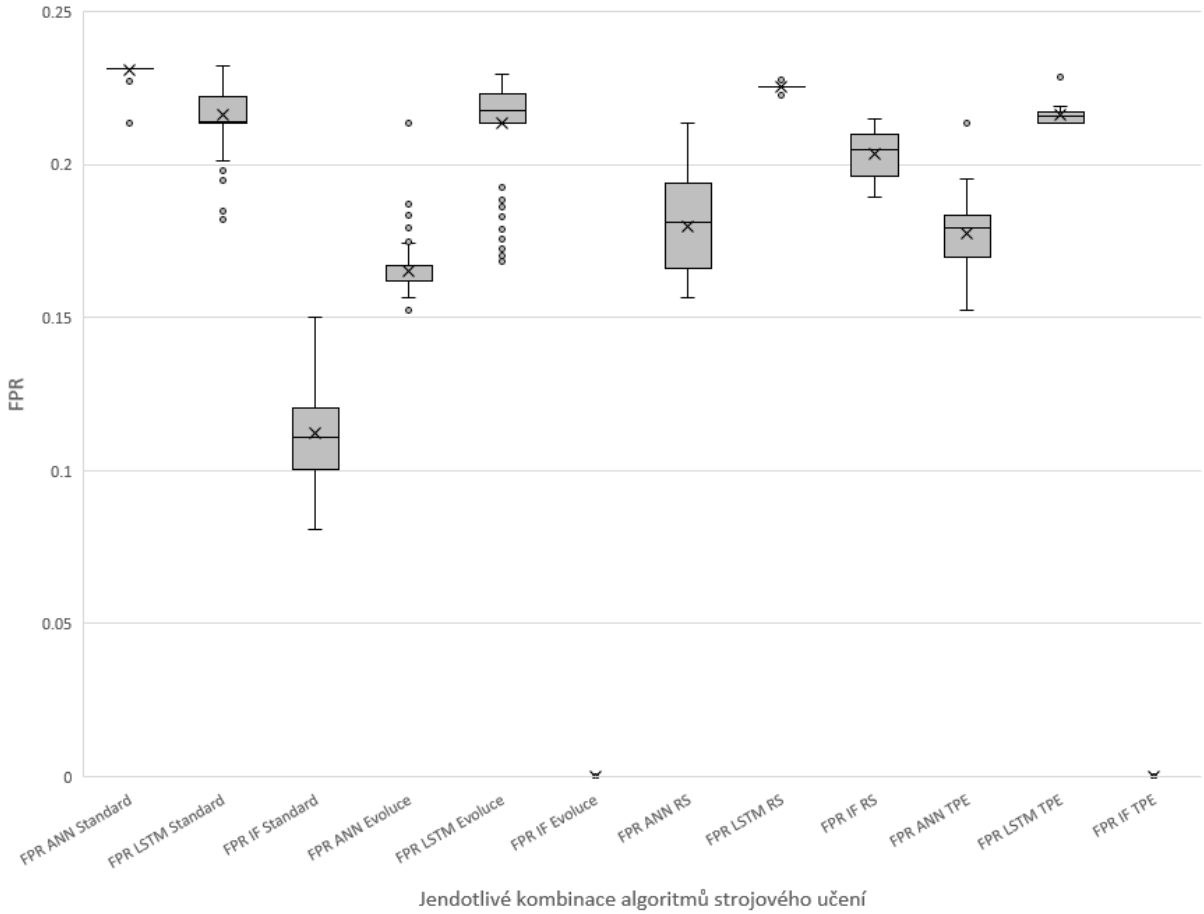
Obr. 122: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_4. [vlastní zdroj]

Na Obr. 123 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_5. Výsledky byly v tomto případě nepřesné (chybné). Jedná se o stejný případ jako v předešlém kybernetickém útoku.



Obr. 123: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_5. [vlastní zdroj]

Na Obr. 124 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_6. Z výsledků lze vybrat dva nejlepší zástupce, kteří se výrazně odlišují od ostatních zástupců. Oba zástupci (IF nastavený podle evolučního algoritmu a IF, který je nastaven podle TPE) dosahují v podstatě stejného výsledku, kde metrika M_{FPR} dosahuje prakticky nulové hodnoty (medián = 0).



Obr. 124: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_6. [vlastní zdroj]

Příloha G: Ověření výsledků algoritmů neuronová síť, LSTM a IF pro finální nastavení hyperparametrů v rámci tří datasetů.

Tab. 123 – Srovnání algoritmů pro detekci anomálií v rámci sedmého a osmého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 2). [vlastní zdroj]

		CA2_7					CA2_8				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.580	0.010	0.583	0.567	3.301	0.480	0.012	0.488	0.460	3.510
	Max	0.591	0.046	0.599	0.635	6.546	0.519	0.079	0.524	0.482	6.996
	Min	0.510	-0.135	0.519	0.528	2.933	0.459	-0.012	0.475	0.423	3.049
LSTM	**Průměr**	0.576	0.007	0.582	0.563	7.129	0.487	0.017	0.491	0.466	7.813
	Max	0.608	0.082	0.614	0.590	28.16	0.536	0.116	0.543	0.490	27.77
	Min	0.555	-0.032	0.565	0.519	5.275	0.440	-0.057	0.451	0.413	5.462
Isolation forest	**Průměr**	0.055	0.033	0.701	0.043	5.524	0.075	0.071	0.773	0.047	5.531
	Max	0.250	0.167	1.000	0.421	8.251	0.354	0.156	1.000	0.518	9.653
	Min	0.000	-0.068	0.007	0.000	4.924	0.000	-0.112	0.000	0.000	4.830

Tab. 124 – Srovnání algoritmů pro detekci anomálií v rámci devátého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 2). [vlastní zdroj]

		CA2_9				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.374	-0.028	0.379	0.398	3.470
	Max	0.429	0.060	0.433	0.421	6.403
	Min	0.323	-0.106	0.330	0.365	2.953
LSTM	**Průměr**	0.395	0.002	0.398	0.390	6.965
	Max	0.437	0.075	0.442	0.402	23.27
	Min	0.373	-0.028	0.379	0.356	5.443
Isolation forest	**Průměr**	0.058	0.046	0.679	0.043	5.516
	Max	0.455	0.145	1.000	0.604	8.238
	Min	0.000	-0.073	0.030	0.000	4.888

Tab. 125 – Srovnání algoritmů pro detekci anomálií v rámci sedmého a osmého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 3). [vlastní zdroj]

		CA3_7					CA3_8				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.510	0.331	0.535	0.166	0.294	0.009	-0.381	0.010	0.386	0.110
	Max	0.517	0.343	0.551	0.196	1.082	0.012	-0.236	0.019	0.401	0.428

	Min	0.492	0.295	0.495	0.152	0.239	0.009	-0.392	0.009	0.187	0.086
LSTM	**Průměr**	0.508	0.325	0.512	0.181	1.349	0.230	-0.039	0.237	0.250	0.158
	Max	0.609	0.469	0.665	0.207	3.222	0.375	0.127	0.376	0.383	0.915
	Min	0.441	0.232	0.444	0.102	1.147	0.011	-0.359	0.012	0.214	0.124
Isolation forest	**Průměr**	0.247	0.324	1.000	0.000	0.530	0.025	0.095	0.999	0.000	0.561
	Max	0.355	0.407	1.000	0.000	1.066	0.033	0.109	1.000	0.000	1.039
	Min	0.150	0.244	0.997	0.000	0.417	0.018	0.081	0.958	0.000	0.417

Tab. 126 – Srovnání algoritmů pro detekci anomálií v rámci devátého kybernetického útoku. Nastavení hyperparametrů podle evolučního algoritmu – (dataset 3). [vlastní zdroj]

		CA3_9				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.274	0.067	0.353	0.168	0.113
	Max	0.281	0.086	0.374	0.214	0.635
	Min	0.257	0.012	0.299	0.152	0.085
LSTM	**Průměr**	0.408	0.188	0.421	0.211	0.157
	Max	0.634	0.491	0.641	0.293	0.826
	Min	0.230	-0.068	0.235	0.140	0.124
Isolation forest	**Průměr**	0.373	0.418	1.000	0.000	0.559
	Max	0.374	0.419	1.000	0.000	1.065
	Min	0.370	0.416	0.998	0.000	0.444

Tab. 127 – Srovnání algoritmů pro detekci anomálií v rámci sedmého a osmého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu RS – (dataset 2). [vlastní zdroj]

		CA2_7					**CA2_8**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.576	0.004	0.580	0.568	7.336	0.507	0.057	0.512	0.445	7.324
	Max	0.625	0.121	0.631	0.608	8.019	0.566	0.171	0.575	0.486	7.973
	Min	0.544	-0.068	0.550	0.499	6.134	0.472	-0.014	0.475	0.373	5.989
LSTM	**Průměr**	0.560	-0.020	0.570	0.570	7.585	0.480	0.008	0.486	0.466	8.193
	Max	0.582	0.021	0.588	0.598	16.93	0.500	0.044	0.505	0.487	22.74
	Min	0.542	-0.060	0.553	0.554	5.940	0.459	-0.033	0.464	0.451	6.258
Isolation forest	**Průměr**	0.281	-0.002	0.576	0.188	12.07	0.235	-0.006	0.476	0.161	12.11
	Max	0.376	0.115	0.686	0.246	16.74	0.290	0.077	0.576	0.205	15.82
	Min	0.208	-0.106	0.468	0.120	9.404	0.179	-0.084	0.391	0.112	9.479

Tab. 128 – Srovnání algoritmů pro detekci anomálií v rámci devátého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu RS – (dataset 2). [vlastní zdroj]

		CA2_9				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.389	-0.006	0.392	0.392	7.303
	Max	0.481	0.150	0.489	0.416	7.948
	Min	0.331	-0.079	0.345	0.324	6.430
LSTM	**Průměr**	0.399	0.012	0.404	0.383	7.460
	Max	0.413	0.034	0.417	0.389	18.55
	Min	0.395	0.004	0.398	0.376	5.794
Isolation forest	**Průměr**	0.226	-0.015	0.379	0.173	12.21
	Max	0.309	0.062	0.458	0.220	17.02
	Min	0.115	-0.121	0.239	0.129	9.510

Tab. 129 – Srovnání algoritmů pro detekci anomálií v rámci sedmého a osmého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu RS – (dataset 3). [vlastní zdroj]

		CA3_7					CA3_8				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.506	0.322	0.524	0.175	0.310	0.010	-0.334	0.013	0.322	0.117
	Max	0.517	0.342	0.548	0.194	1.031	0.013	-0.233	0.020	0.399	0.565
	Min	0.498	0.302	0.500	0.157	0.227	0.009	-0.391	0.009	0.185	0.088
LSTM	**Průměr**	0.501	0.316	0.506	0.182	0.761	0.019	-0.265	0.025	0.239	0.097
	Max	0.508	0.328	0.522	0.189	3.191	0.023	-0.252	0.031	0.276	0.676
	Min	0.488	0.299	0.494	0.170	0.576	0.015	-0.291	0.019	0.225	0.069
Isolation forest	**Průměr**	0.482	0.279	0.482	0.203	0.373	0.111	-0.139	0.148	0.203	0.390
	Max	0.500	0.307	0.505	0.215	0.804	0.199	-0.034	0.256	0.216	0.774
	Min	0.470	0.260	0.466	0.190	0.290	0.038	-0.220	0.056	0.190	0.304

Tab. 130 – Srovnání algoritmů pro detekci anomálií v rámci devátého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu RS – (dataset 3). [vlastní zdroj]

		CA3_9				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.271	0.056	0.341	0.176	0.118
	Max	0.279	0.081	0.367	0.214	0.643
	Min	0.257	0.012	0.299	0.157	0.086
LSTM	**Průměr**	0.251	-0.009	0.277	0.238	0.096
	Max	0.257	0.005	0.289	0.262	0.644
	Min	0.244	-0.032	0.260	0.225	0.071
Isolation forest	**Průměr**	0.269	0.033	0.317	0.204	0.399

	Max	0.275	0.053	0.336	0.216	0.742
	Min	0.264	0.019	0.305	0.186	0.306

Tab. 131 – Srovnání algoritmů pro detekci anomálií v rámci sedmého a osmého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu TPE – (dataset 2). [vlastní zdroj]

		CA2_7					**CA2_8**				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**	**M_{F1}**	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.571	-0.009	0.575	0.575	4.962	0.469	-0.016	0.473	0.481	4.990
	Max	0.639	0.146	0.641	0.626	15.44	0.531	0.105	0.537	0.565	14.997
	Min	0.527	-0.103	0.534	0.490	3.846	0.380	-0.188	0.383	0.420	3.981
LSTM	**Průměr**	0.588	0.040	0.596	0.540	12.505	0.509	0.060	0.514	0.444	12.473
	Max	0.594	0.055	0.603	0.565	14.654	0.523	0.084	0.526	0.467	14.436
	Min	0.567	0.001	0.579	0.531	9.923	0.484	0.013	0.489	0.435	10.774
Isolation forest	**Průměr**	0.266	0.000	0.578	0.174	33.547	0.226	-0.005	0.477	0.152	33.376
	Max	0.318	0.075	0.663	0.220	49.587	0.266	0.057	0.552	0.203	46.802
	Min	0.188	-0.066	0.497	0.120	26.948	0.187	-0.062	0.413	0.111	26.880

Tab. 132 – Srovnání algoritmů pro detekci anomálií v rámci devátého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu TPE – (dataset 2). [vlastní zdroj]

		CA2_9				
		M_{F1}	**M_{MCC}**	**M_{Prec}**	**M_{FPR}**	**Čas**
Neuronová síť	**Průměr**	0.388	-0.008	0.391	0.393	4.937
	Max	0.448	0.088	0.449	0.438	15.86
	Min	0.322	-0.115	0.325	0.332	3.986
LSTM	**Průměr**	0.392	-0.003	0.394	0.393	12.561
	Max	0.398	0.007	0.400	0.397	14.872
	Min	0.385	-0.014	0.388	0.389	10.769
Isolation forest	**Průměr**	0.202	-0.026	0.366	0.159	33.404
	Max	0.256	0.040	0.441	0.202	48.478
	Min	0.142	-0.087	0.296	0.111	26.915

Tab. 133 – Srovnání algoritmů pro detekci anomálií v rámci sedmého a osmého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu TPE – (dataset 3). [vlastní zdroj]

		CA3_7					CA3_8				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas	M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.506	0.320	0.521	0.178	0.237	0.010	-0.348	0.012	0.340	0.099
	Max	0.520	0.344	0.547	0.194	1.058	0.015	-0.222	0.023	0.400	0.914
	Min	0.494	0.302	0.500	0.157	0.174	0.009	-0.392	0.009	0.171	0.071
LSTM	**Průměr**	0.495	0.306	0.497	0.188	1.406	0.164	-0.097	0.169	0.230	0.158
	Max	0.509	0.330	0.519	0.192	4.558	0.358	0.106	0.361	0.251	0.899
	Min	0.490	0.298	0.490	0.174	0.900	0.012	-0.254	0.017	0.214	0.102
Isolation forest	**Průměr**	0.240	0.319	1.000	0.000	0.428	0.025	0.094	1.000	0.000	0.467
	Max	0.318	0.379	1.000	0.000	0.844	0.033	0.109	1.000	0.000	0.898
	Min	0.133	0.229	0.995	0.000	0.348	0.020	0.085	0.962	0.000	0.380

Tab. 134 – Srovnání algoritmů pro detekci anomálií v rámci devátého kybernetického útoku. Nastavení hyperparametrů podle optimalizačního algoritmu TPE – (dataset 3). [vlastní zdroj]

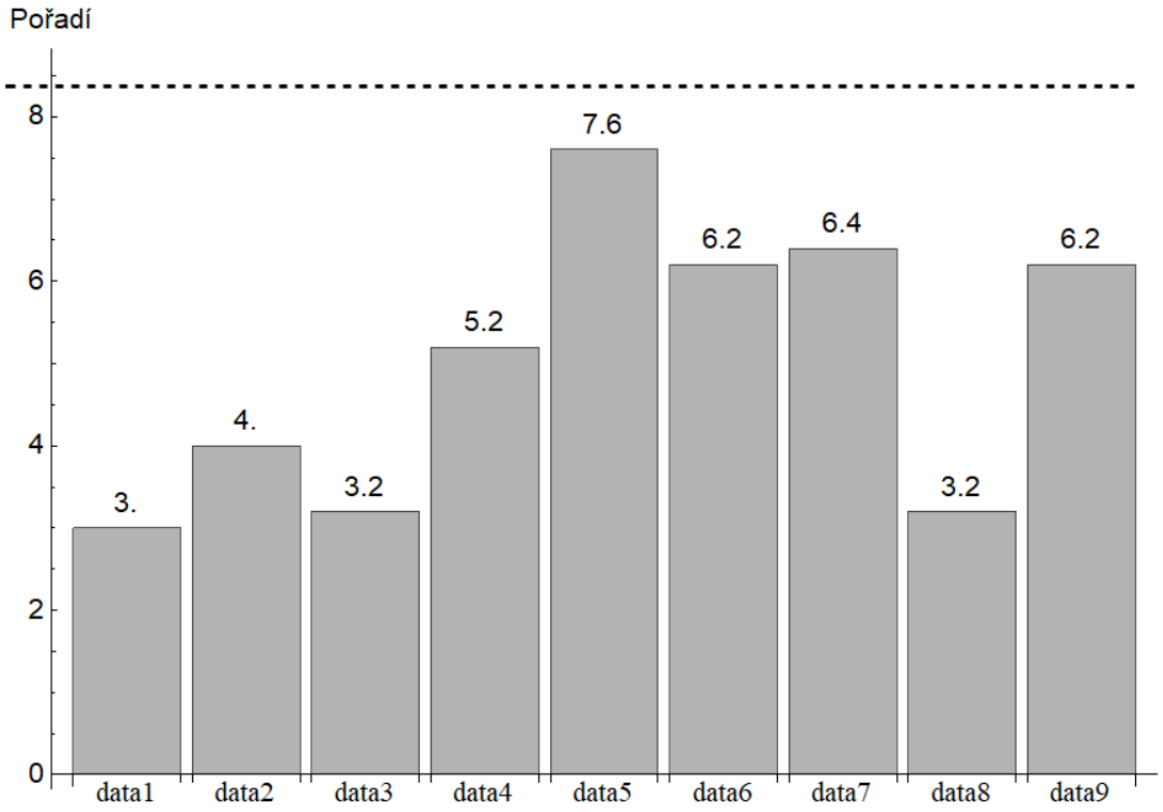
		CA3_9				
		M_{F1}	M_{MCC}	M_{Prec}	M_{FPR}	Čas
Neuronová síť	**Průměr**	0.270	0.053	0.338	0.179	0.089
	Max	0.279	0.081	0.368	0.199	0.432
	Min	0.262	0.029	0.314	0.157	0.071
LSTM	**Průměr**	0.344	0.112	0.366	0.215	0.153
	Max	0.455	0.239	0.456	0.228	0.861
	Min	0.257	0.014	0.297	0.214	0.103
Isolation forest	**Průměr**	0.372	0.417	1.000	0.000	0.450
	Max	0.374	0.419	1.000	0.000	0.859
	Min	0.370	0.416	0.998	0.000	0.375

Ověření výsledků algoritmů – dataset 2

Tab. 135 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_7 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.50759	0.03074

Podle Tab. 135 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



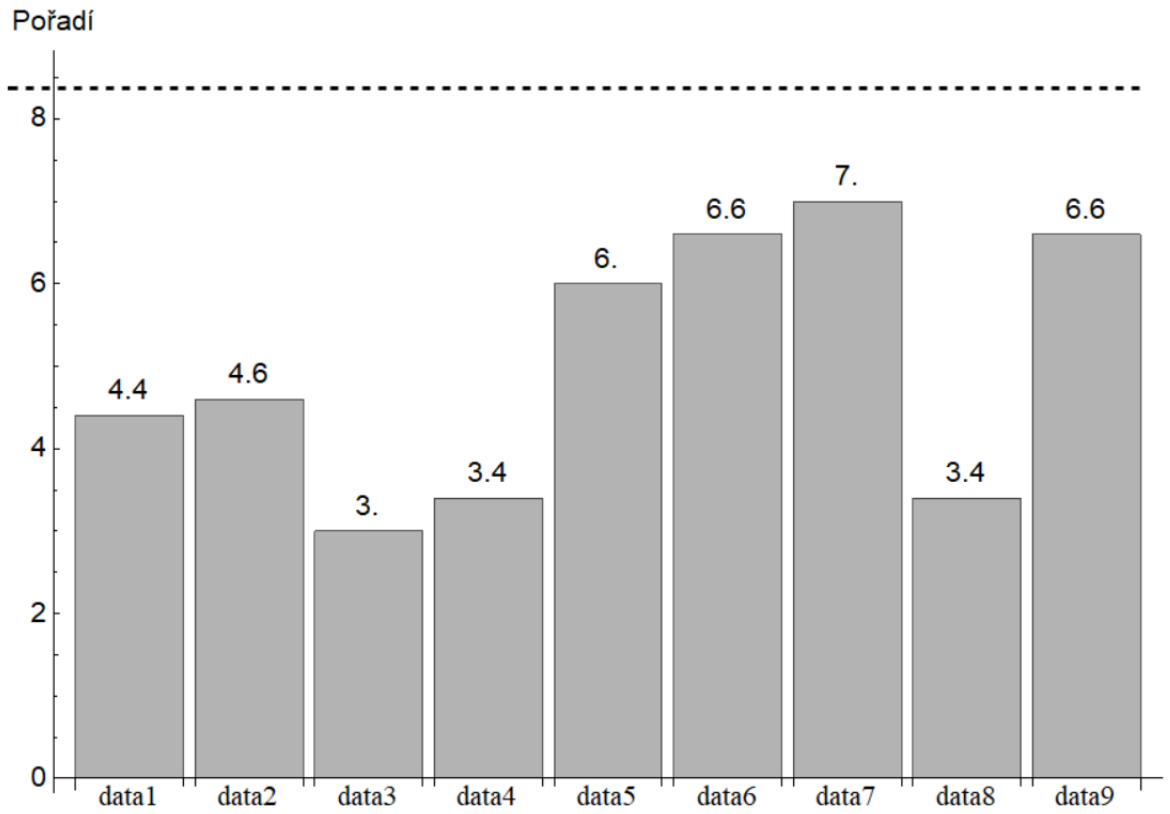
Obr. 125: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_7. [vlastní zdroj]

Na Obr. 125 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku neexistuje algoritmu strojového učení, který by se významně negativně odlišoval od ostatních algoritmů. Nejblíže k tomu má algoritmus LSTM nastavený pomocí RS algoritmu (data5). Oproti tomu lze identifikovat neuronovou síť nastavenou pomocí evolučního algoritmu (data1), IF nastavený podle evolučního algoritmu (data3) a algoritmus LSTM nastavený podle TPE (data 8) jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 136 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_8 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.96421	0.08405

PodleTab. 136 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



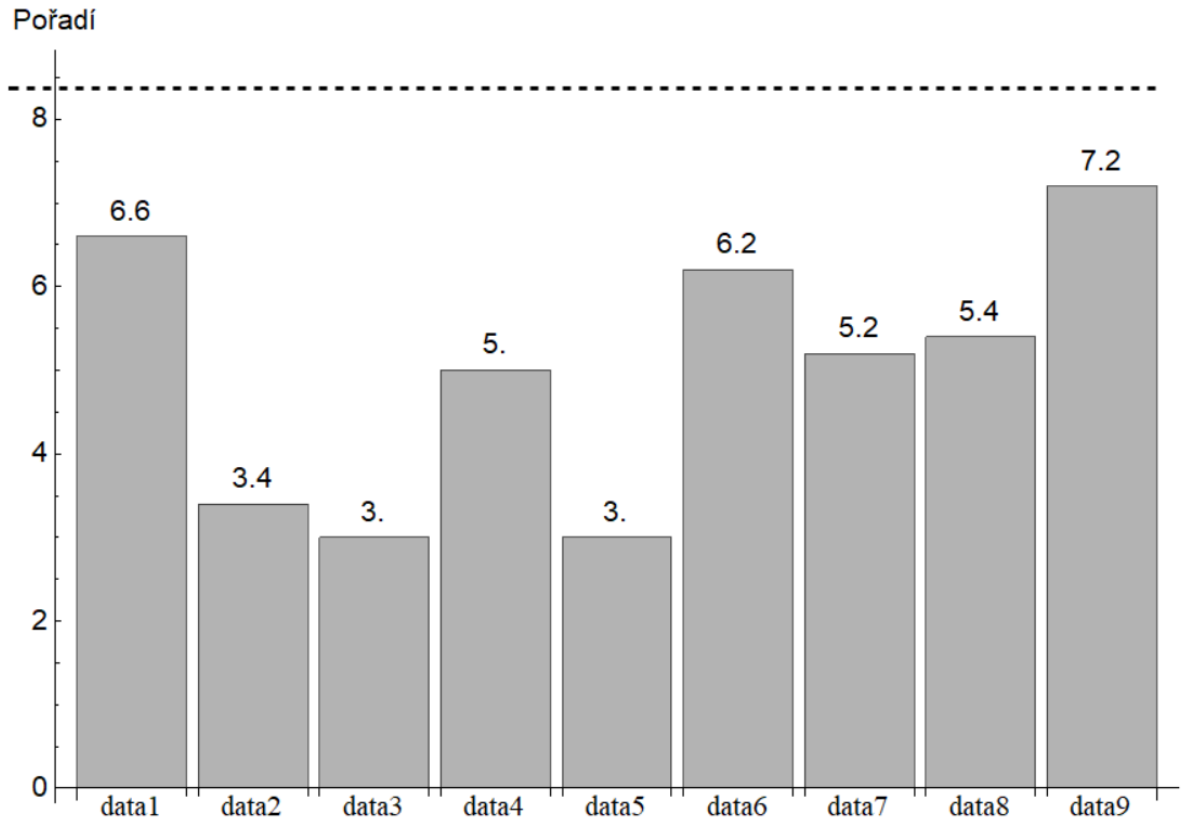
Obr. 126: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_8. [vlastní zdroj]

Na Obr. 126 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Tab. 137 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA2_9 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	1.94059	0.0878

Podle Tab. 137 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



Obr. 127: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA2_9. [vlastní zdroj]

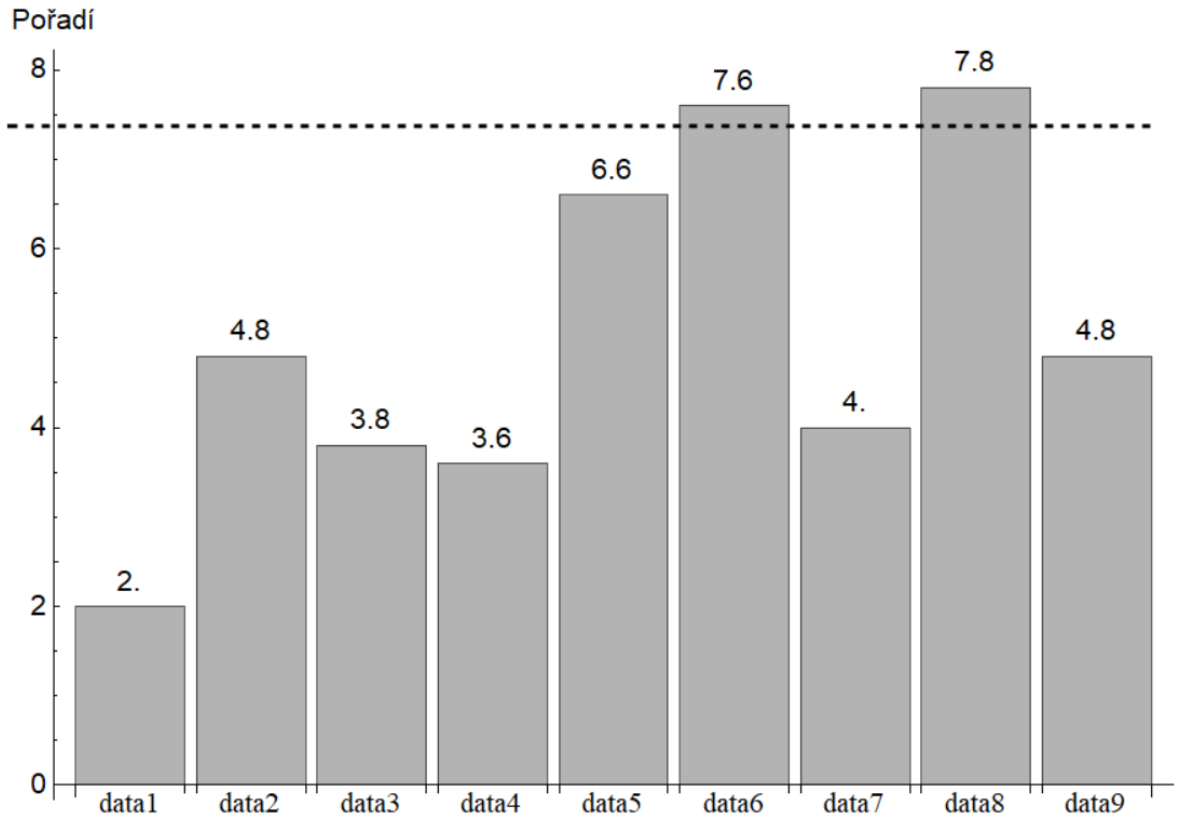
Na Obr. 127 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Ověření výsledků algoritmů – dataset 3

Tab. 138 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_7 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	4.17439	0.00166

Podle Tab. 138 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



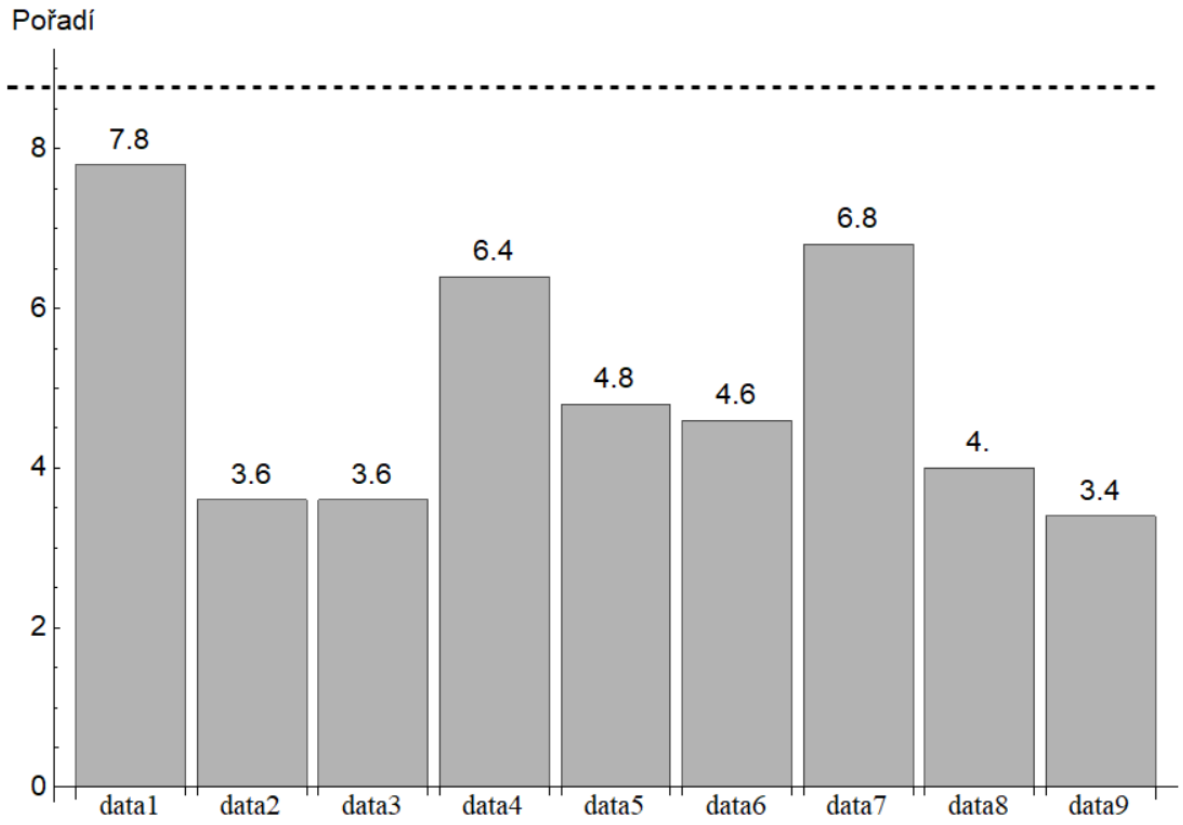
Obr. 128: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_7. [vlastní zdroj]

Na Obr. 128 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku existují algoritmy strojového učení, které se významně negativně odlišují od ostatních algoritmů. Jedná se o algoritmus strojového učení IF, který je nastavený podle optimalizačního algoritmu RS (data6) a algoritmus LSTM, který je nastaven podle TPE (data8). Oproti tomu, lze identifikovat neuronovou síť nastavenou pomocí evolučního algoritmu (data1) a neuronovou síť nastavenou pomocí algoritmu RS (data4), jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Tab. 139 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_8 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.10998	0.06415

Podle Tab. 139 není nulová hypotéza zamítnuta. P-hodnota je vyšší než 5 %, a tudíž mezi daty neexistuje statisticky významný rozdíl.



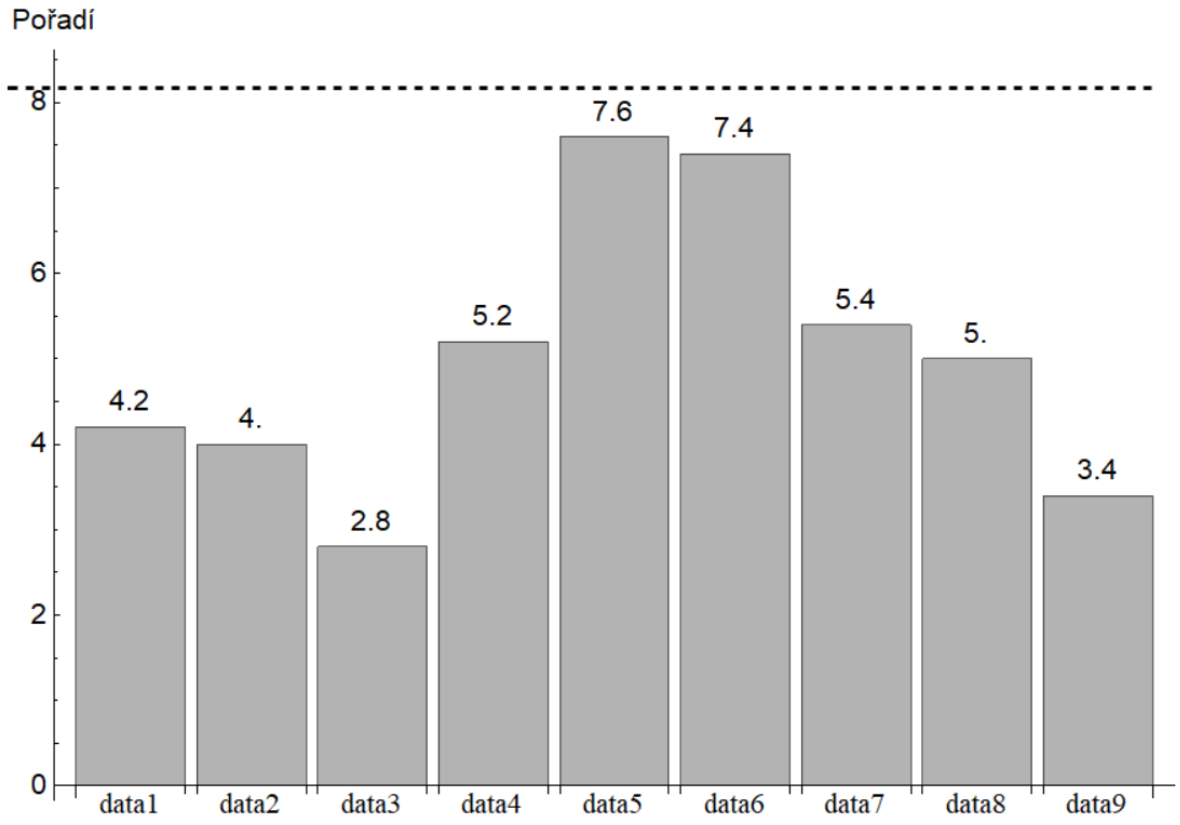
Obr. 129: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_8. [vlastní zdroj]

Na Obr. 129 je proveden Friedmanův test pro určení pořadí jednotlivých kombinací technik pro úpravu dat na základě hodnotících kritérií. Mezi jednotlivými alternativami zpracování dat není významnější rozdíl v rámci tohoto kybernetického útoku.

Tab. 140 – Výsledky statistiky pro Friedmanův test v rámci kybernetického útoku – CA3_9 – testování všech algoritmů. [vlastní zdroj]

	Testovací statistika	p-hodnota
Friedmanův test	2.27615	0.04714

Podle Tab. 140 je nulová hypotéza zamítnuta. P-hodnota je nižší než 5 %, a tudíž mezi daty existuje statisticky významný rozdíl.



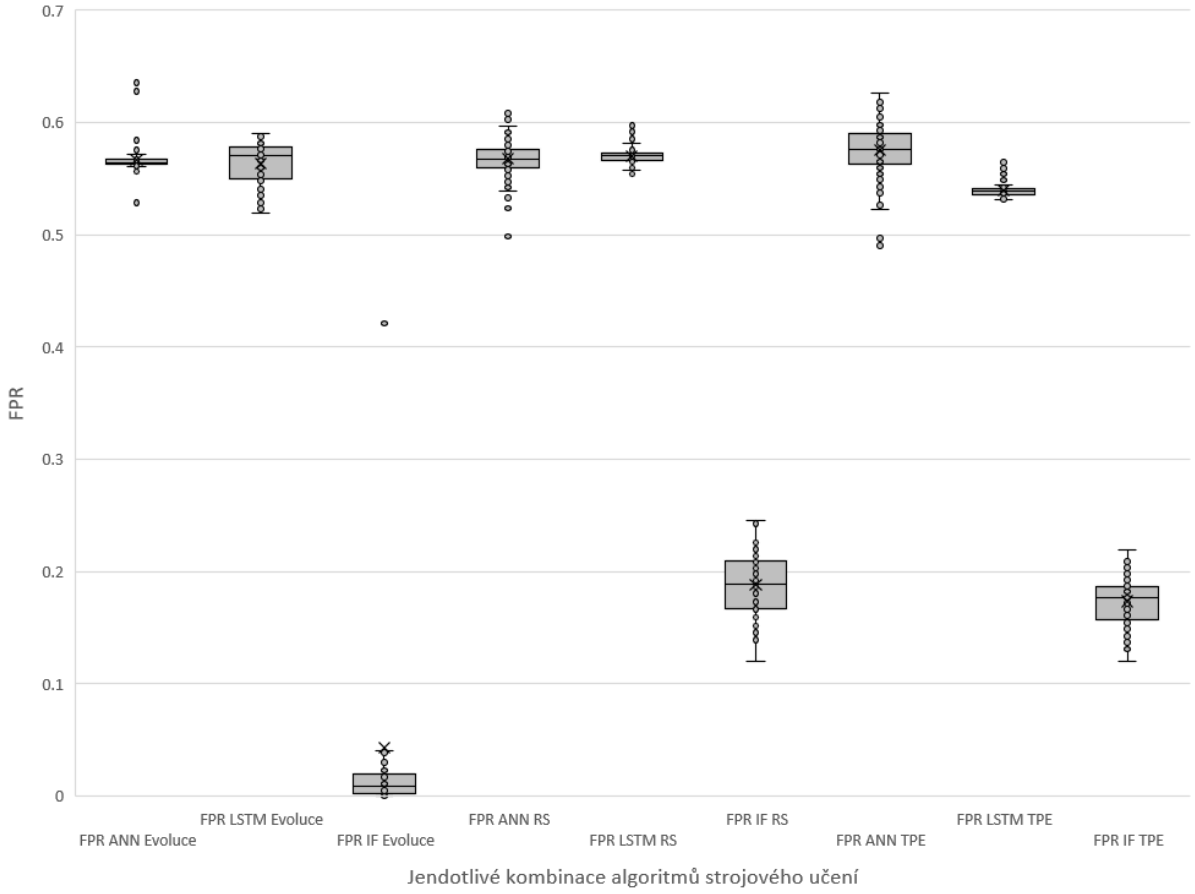
Obr. 130: Friedmanův test včetně Nemenyiho kritické vzdálenosti pro různá nastavení algoritmů strojového učení v rámci kybernetického útoku – CA3_9. [vlastní zdroj]

Na Obr. 130 je proveden Friedmanův test pro algoritmy strojového učení (neuronová síť, LSTM, IF) pro nastavení hyperparametrů podle optimalizačních algoritmů. V rámci tohoto kybernetického útoku neexistuje algoritmus strojového učení, který by se významně negativně odlišoval od ostatních algoritmů. Nejblíže k tomu má algoritmus LSTM nastavený pomocí RS algoritmu (data5). Oproti tomu, lze identifikovat algoritmus IF, který je nastavený pomocí evolučního algoritmu (data3) a algoritmus IF, který je nastaven pomocí algoritmu TPE (data9), jako zástupce s nejlepšími detekčními schopnostmi z množiny algoritmů.

Příloha H: Porovnání metriky M_{FPR} pro jednotlivá řešení v rámci jejich ověření.

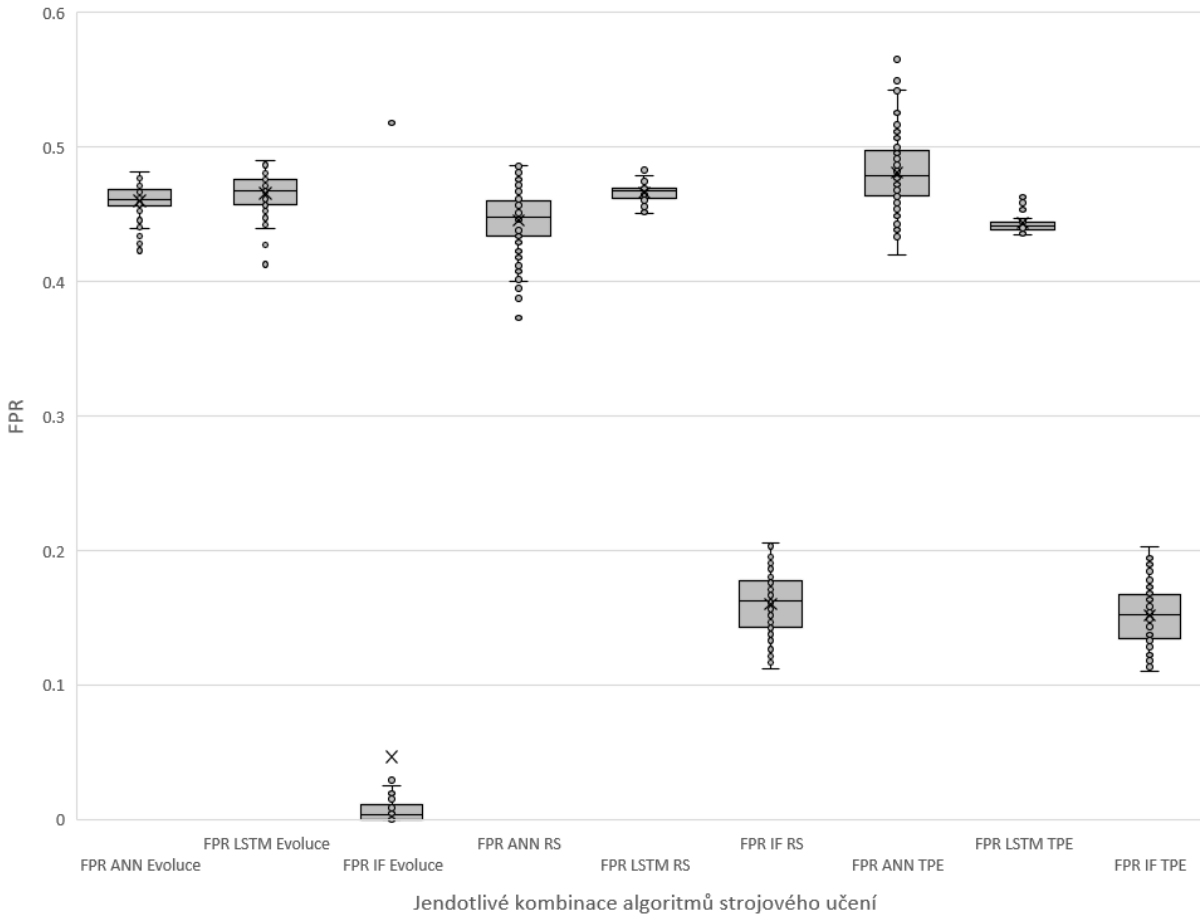
Porovnání metriky M_{FPR} pro jednotlivá řešení v rámci jejich ověření – dataset 2

Na Obr. 131 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_7. Z výsledků lze identifikovat nejlepšího zástupce, který se odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.008).



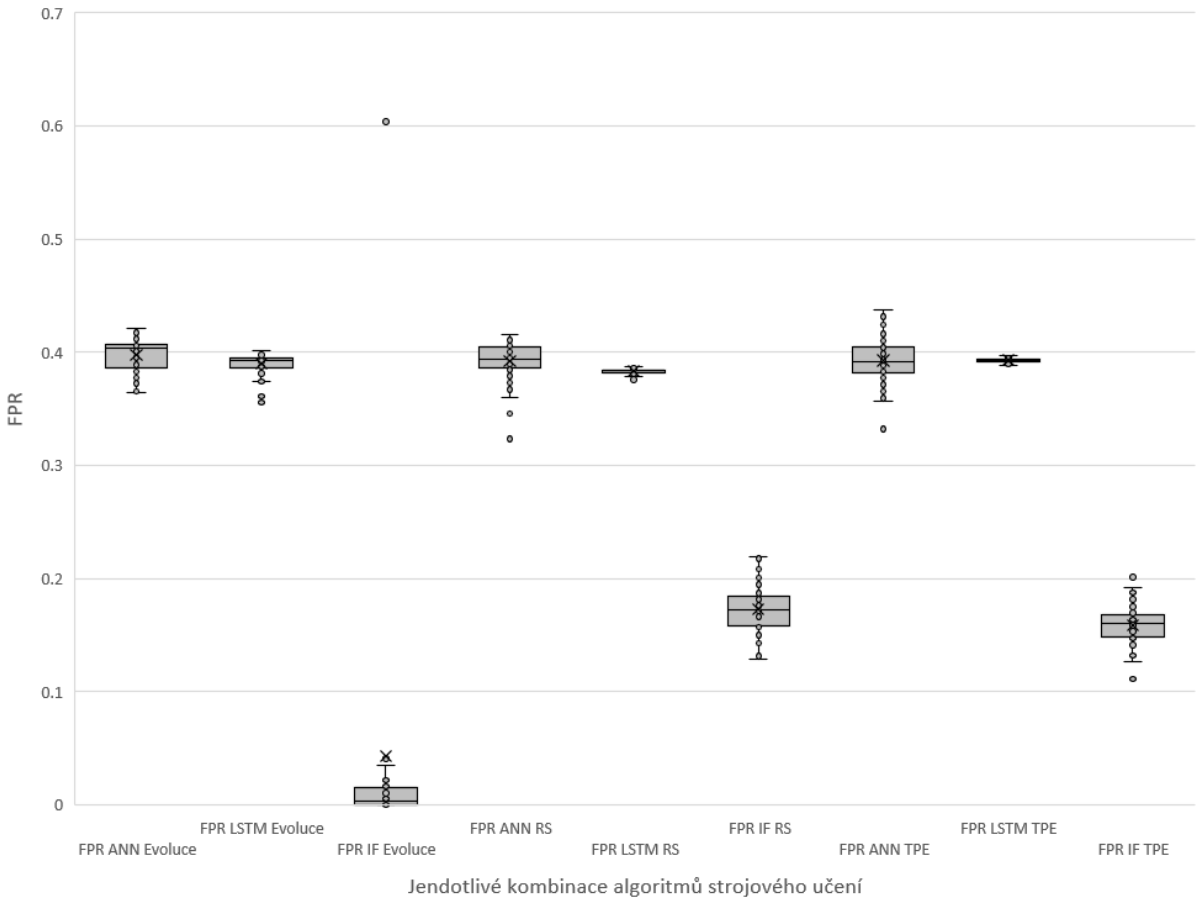
Obr. 131: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_7. [vlastní zdroj]

Na Obr. 132 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_8. Z výsledků lze identifikovat nejlepšího zástupce, který se odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.003).



Obr. 132: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_8. [vlastní zdroj]

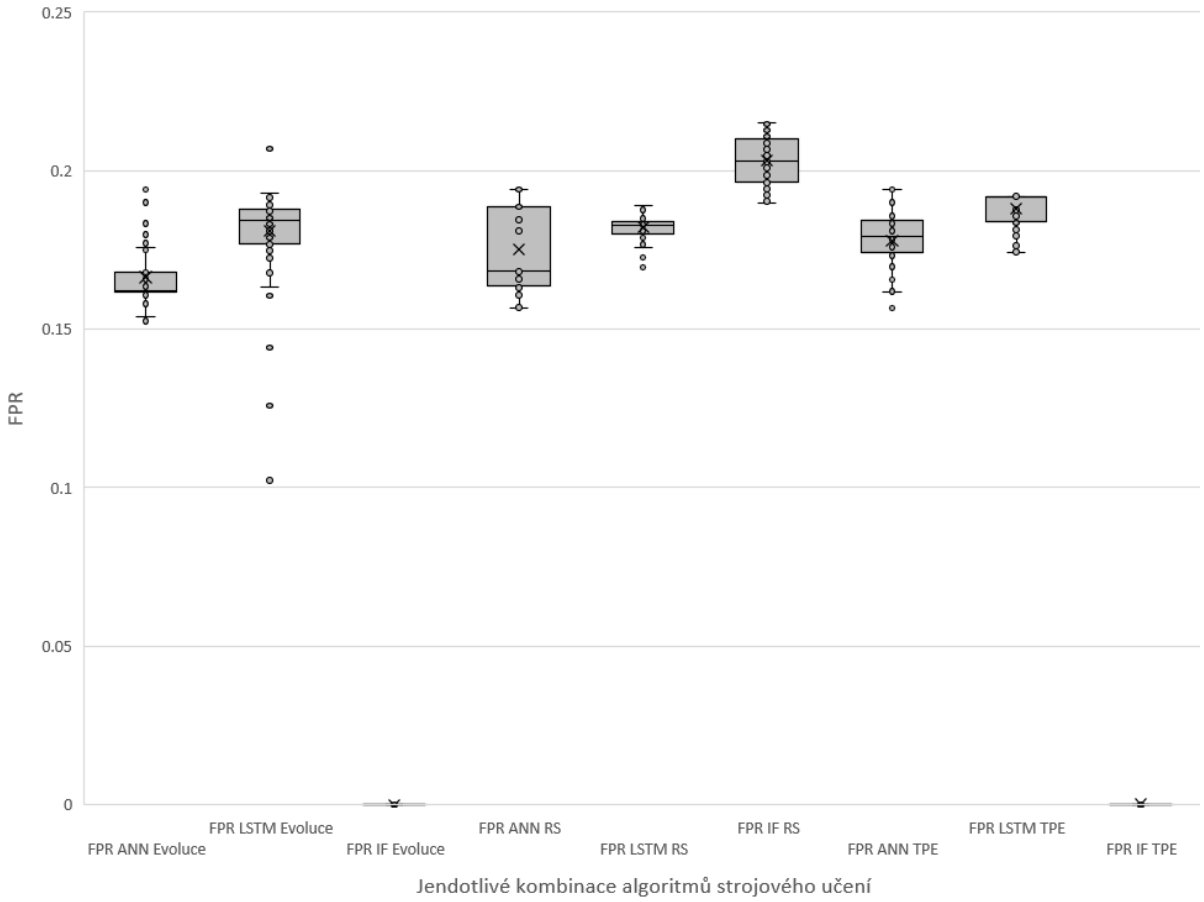
Na Obr. 133 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA2_9. Z výsledků lze identifikovat nejlepšího zástupce, který se odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0.004).



Obr. 133: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA2_9. [vlastní zdroj]

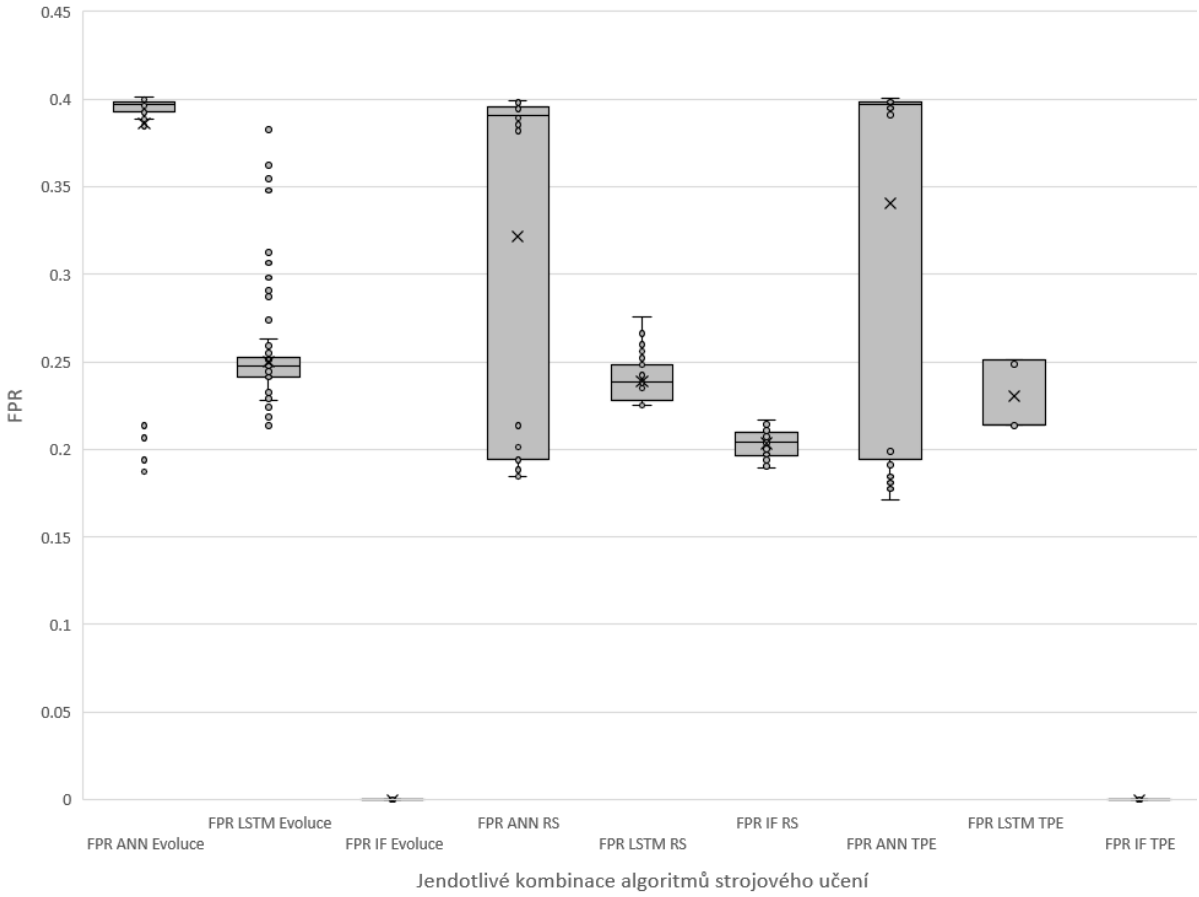
Porovnání metriky M_{FPR} pro jednotlivá řešení v rámci jejich ověření – dataset 3

Na Obr. 134 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_7. Z výsledků lze identifikovat nejlepšího zástupce, který se odlišuje od ostatních zástupců. Nejlepším zástupcem je algoritmus IF nastavený pomocí evolučního algoritmu (medián: 0).



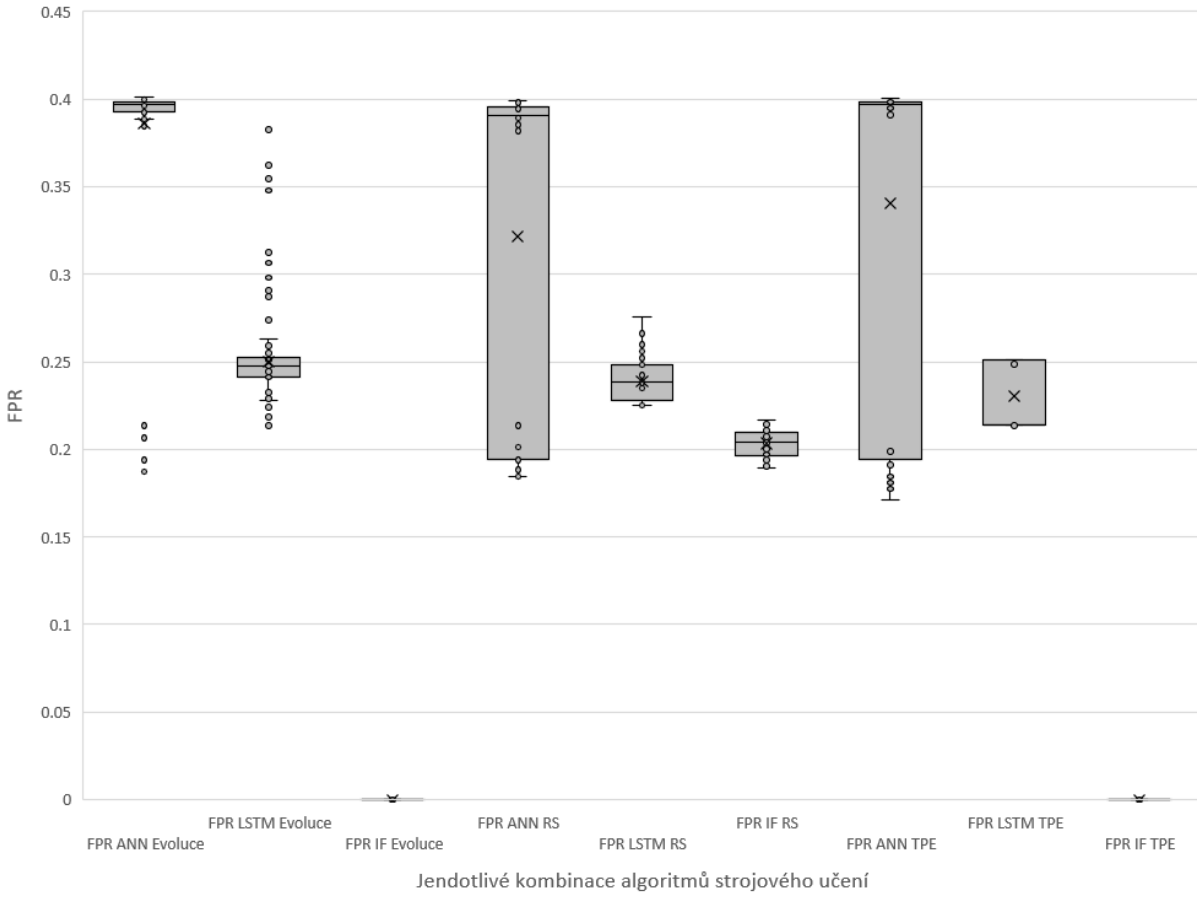
Obr. 134: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_7. [vlastní zdroj]

Na Obr. 135 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_8. Z výsledků lze identifikovat dva nejlepší zástupce, kteří se odlišují od ostatních zástupců. Těmito zástupci jsou: algoritmus IF nastavený pomocí evolučního algoritmu, IF algoritmus nastavený podle TPE (medián: 0).



Obr. 135: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_8. [vlastní zdroj]

Na Obr. 136 jsou zobrazeny výsledky metriky M_{FPR} pro jednotlivé zástupce v rámci kybernetického útoku CA3_9. Z výsledků lze identifikovat dva nejlepší zástupce, kteří se odlišují od ostatních zástupců. Těmito zástupci jsou: algoritmus IF nastavený pomocí evolučního algoritmu, IF algoritmus nastavený podle TPE (medián: 0).



Obr. 136: Porovnání metriky M_{FPR} pro jednotlivé kombinace algoritmu strojového učení a optimalizačních algoritmů pro kybernetický útok – CA3_9. [vlastní zdroj]

14. PUBLIKAČNÍ AKTIVITY AUTORA

Mezinárodní publikace:

1) VÁVRA, Jan; HROMADA, Martin. An evaluation of cyber threats to industrial control systems. In: International Conference on Military Technologies (ICMT) 2015. IEEE, 2015. p. 1-5. ISBN 978-80-7231-976-3.
2) VÁVRA, Jan; HROMADA, Martin; JAŠEK, Roman. Specification of the current state vulnerabilities related to industrial control systems. International Journal of Online and Biomedical Engineering (iJOE), 2015, 11.5: 64-68. ISSN 1868-1646.
3) VÁVRA, Jan; HROMADA, Martin. Comparison of the intrusion detection system rules in relation with the SCADA systems. In: Computer Science On-line Conference. Springer, Cham, 2016. p. 159-169. ISBN 978-3-319-33620-6.
4) VÁVRA, Jan; HROMADA, Martin. Possibilities of the Search Engine Shodan in Relation to SCADA. In: SECURWARE 2016, The Tenth International Conference on Emerging Security Information, Systems and Technologies, 2016. p. 130-135. IARIA. ISBN 978-1-61208-493-0.
5) VÁVRA, Jan; HROMADA, Martin. Determination of optimal cluster number in connection to SCADA. In: Computer Science On-line Conference. Springer, Cham, 2017. p. 136-147. ISBN 978-3-319-57141-6.
6) VÁVRA, Jan; HROMADA, Martin. Evaluation of anomaly detection based on classification in relation to SCADA. In: 2017 International Conference on Military Technologies (ICMT). IEEE, 2017. p. 330-334. ISBN 978-1-5386-1988-9.
7) VÁVRA, Jan; HROMADA, Martin. Anomaly detection system based on classifier fusion in ics environment. In: 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT). IEEE, 2017. p. 32-38. ISBN 978-1-4673-9899-2.
8) VÁVRA, Jan; HROMADA, Martin. Novelty Detection System Based on Multi-criteria Evaluation in Respect of Industrial Control System. In: Computer Science On-line Conference. Springer, Cham, 2018. p. 280-289. ISBN 978-331991191-5.
9) VAVRA, Jan; HROMADA, Martin. Comparative Study of Feature Selection Techniques Respecting Novelty Detection in the Industrial Control System Environment. Annals of DAAAM and Proceedings of the

International DAAAM Symposium, 2018, 29. p. 1084-1091. ISBN 978-3-902734-20-4.
10) VAVRA, Jan; HROMADA, Martin. Optimization of the Novelty Detection Model Based on LSTM Autoencoder for ICS Environment. In: Proceedings of the Computational Methods in Systems and Software. Springer, Cham, 2019. p. 306-319. ISBN 978-3-030-30328-0.
11) VAVRA, Jan; HROMADA, Martin. Evaluation of Data Preprocessing Techniques for Anomaly Detection Systems in Industrial Control System. Annals of DAAAM & Proceedings, 2019, 30. p. 738-745.

Tuzemské publikace:

1) VAVRA, Jan. Optimization of Crisis Management in Municipality via GIS. In: Trilobit [online]. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2015. č. 1/2015, ISSN 1804-1795.
2) VAVRA, Jan. Ochrana ICT před škodlivým působením blesku. In: Trilobit [online]. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2015. č. 1/2015, ISSN 1804-1795.
3) VAVRA, Jan; HROMADA, Martin. Specifikace Kybernetických Incidentů Vztahujících se k ICS. In: Bezpečnostní technologie, systémy a management 2015: Sborník příspěvků 5. mezinárodní konference. 1. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2015, s. 1-6. ISBN 978-80-7454-559-7.
4) VAVRA, Jan; HROMADA, Martin. Zhodnocení Detekčních Metodologií IDS ve Vztahu k ICS. In Sborník příspěvků z mezinárodní konference MLADÁ VĚDA 2016. Ostrava: Sdružení požárního a bezpečnostního inženýrství, z.s., 2016, s. 466-471. ISBN 978-80-7385-177-4.
5) VAVRA, Jan; HROMADA, Martin. Umělá inteligence jako nástroj ochrany kritické infrastruktury. In Sborník příspěvků z mezinárodní konference MLADÁ VĚDA 2019. Ostrava: Sdružení požárního a bezpečnostního inženýrství, z.s., 2019, s. 89- 99. ISBN 978-80-7385-222-1.
6) VAVRA, Jan; HROMADA, Martin. Metodika pro výběr metod určených pro kvantifikaci penalizačních faktorů v oblasti konvergované bezpečnosti. In: Trilobit [online]. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2019. č. 3/2019, ISSN 1804-1795.

15. ODBORNÝ ŽIVOTOPIS AUTORA

Ing. Jan Vávra

Osobní údaje
Datum narození: 13. 10. 1987

Adresa: Úprkova 1809, 68603 Staré Město (Česká republika)

E-mail: jvavra@utb.cz

Tel.: +420722691886

Vzdělání

Dosažené vzdělání:
vysokoškolské II. stupě (Magisterské) – Ing.

09/2014–do současnosti
Ph.D. student ve studijním oboru Inženýrská informatika
Univerzita Tomáše Bati ve Zlíně, Zlín (Česká republika)

17/06/2014–18/06/2014
Certifikát z oblasti základů elektronického zabezpečení objektů JABLOTRON ALARMS a.s.,
Zlín (Česká republika)

09/2009–06/2014
Inženýrský titul v oboru Bezpečnostní technologie, systémy a management
Univerzita Tomáše Bati, Zlín (Česká republika)

09/2004–05/2008
Maturita v oboru Technické lyceum
Střední průmyslová škola, Uherské Hradiště (Česká republika)

Přehled aktivit během studia

1. 1. 2019 - 31. 12. 2019
Řešitel projektu IGA (IGA/FAI/2019/002) Detekce kybernetických útoků v prostředí průmyslových řídicích systémů prostřednictvím strojového učení.

1. 6. 2018 - 31. 7. 2018
Pracovní stáž v zahraničí Erasmus+ - Holandsko - University of Twente.

1. 1. 2018 - 31. 12. 2018
Řešitel projektu IGA (IGA/FAI/2018/003) Konceptuální návrh metodiky detekce anomálií vztahující se k průmyslovým řídicím systémům.

1. 1. 2017 - 31. 12. 2017
Řešitel projektu IGA (IGA/FAI/2017/003) Evaluace detekčních metodologií ve vztahu ke SCADA systémům.

1. 6. 2016 - 31. 7. 2016
Pracovní stáž v zahraničí Erasmus+ - Itálie - University of Cagliari.

1. 1. 2016 - 31. 12. 2016
Řešitel projektu (IGA IGA/FAI/2016/014) - Specifikace ICS kybernetické bezpečnosti se zaměřením na IDPS.

od 2016 do 2019
Spoluřešitel projektu ev. no. VI20172019054 - Analytický programový modul pro hodnocení odolnosti v reálném čase z hlediska konvergované bezpečnosti.

1. 3. 2015 - 31. 12. 2015
Řešitel projektu IGA (IGA/FAI/2015/042) - Analýza kybernetické bezpečnosti v organizaci se zaměřením na ICS.

20. 9. 2015 - 20. 12. 2015
Studijní zahraniční pobyt Erasmus+ - Řecko - University of Peloponnese.

od 2015 do 2018
Spoluřešitel projektu ev. no. VI20152019049 "RESILIENCE 2015: Dynamické hodnocení odolnosti souvztažných subsystémů kritické infrastruktury.

Pedagogická činnost

Základy informatiky – A1ZIN – prezenční studium.
Informatika – A1INF – prezenční studium.

Ing. Jan Vávra

Návrh a ověření systému detekce anomálií založeného na strojovém učení v průmyslových řídicích systémech

Design and verification of anomaly detection system based on machine learning in industrial control systems

Disertační práce

Sazba: Ing. Jan Vávra

Publikace neprošla jazykovou ani redakční úpravou.

Rok vydání 2020