SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, PhD. Akhmedov Ramis

« 04 » 06 2024

Topic of the thesis:
Intrusion detection system with deep packet inspection

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science", SDU University

Head of Department	Zhanar Mukash
Academic Supervisor	Rashid Baimukashev
Master student	Farikha Nauruzbayeva

Kaskelen, 2024

Ministry of Science and Higher Education of the Republic of Kazakhstan

SDU University

Nauruzbayeva Farikha

Master Thesis: "Intrusion detection system with deep packet inspection"

THESIS

Presented in Partial Fulfilment for the

Degree of Master of Technical Science in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Rashid Baimukashev**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science / Department of Mathematics and Natural Sciences

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, Ph.D. Akhmedov Ramis

_____

« _____» _____ 2024

Topic of the thesis:

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science" / "7M05401-Mathematics", SDU University, 2022-2024

Head of Department _____

Academic Supervisor _____

Master student _____

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Nauruzbayeva Farikha

September 2024

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Rashid Baimukashev, for his invaluable guidance and support throughout my research journey. I am truly thankful for his **patience**, insightful advice, and unwavering encouragement. Furthermore, I extend my appreciation for his understanding of my area of interest, allowing me to confidently pursue the topic "Intrusion Detection System with Deep Packet Inspection" without any fear and help me to choose dataset, algorithms and etc. Thanks to my supervisor for **finishing the university**. Supervisor's mentorship has been instrumental in shaping my academic and professional growth, and I am truly fortunate to have had the opportunity to work under his supervision.

Dedication

This thesis is dedicated to:

All future researchers in cybersecurity sphere. Never give up!

Abstract

The rapid expansion of the internet has revolutionized modern life, yet it also poses significant risks due to vulnerabilities in increasingly complex software and networks. Addressing these risks requires robust methods for preventing attacks. Firewalls serve as critical security tools in network protection, distinguishing between secure and less secure networks and regulating information flow. However, they alone cannot fully safeguard against malicious activities. Intrusion Detection Systems (IDS), particularly those incorporating Deep Packet Inspection (DPI), have emerged as effective means of identifying and mitigating network threats.
The research aims to evaluate the effectiveness of DPI algorithms in detecting and responding to network intrusions. By reviewing existing literature and conducting experiments, the study seeks to identify the most efficient algorithms based on speed and accuracy metrics.
The research objectives include proposing algorithm designs to enhance IDS performance and developing an intrusion detection system algorithm. The methodology involves studying various DPI algorithms, normalizing datasets, and comparing algorithm performance through experimentation. Preliminary results from algorithms like K-Nearest demonstrate ongoing progress in algorithm development.
While the project faces challenges such as research environment limitations, the commitment to understanding the subject thoroughly remains unwavering. By leveraging Linux operating systems and diligent research, the study aims to contribute valuable insights into intrusion detection system effectiveness, potentially leading to the development of more efficient algorithms.

Аңдатпа

Интернеттің жылдам таралуы қазіргі өмірде төңкеріс жасады, бірақ ол сонымен бірге барған сайын күрделі бағдарламалық қамтамасыз ету мен желілердегі осалдықтарға байланысты айтарлықтай қауіп төндіреді. Бұл тәуекелдерді жою шабуылдың алдын алудың сенімді әдістерін қажет етеді. Желіаралық қалқандар желіні қорғауда, қауіпсіз және аз қорғалған желілерді ажыратуда және ақпарат ағынын реттеуде маңызды қауіпсіздік құралдары ретінде қызмет етеді. Алайда, олар өз бетімен зиянды әрекеттерден толық қорғай алмайды. Интрузияны анықтау жүйелері (IDS), әсіресе терең пакеттік тексеруді (DPI) қамтитын жүйелер желілік қауіптерді анықтаудың және азайтудың тиімді құралы болды.

Зерттеудің мақсаты желілік енулерді анықтау және оларға жауап берудегі DPI алгоритмдерінің тиімділігін бағалау болып табылады. Қолданыстағы әдебиеттерді шолу және эксперименттер жүргізу арқылы зерттеу жылдамдық пен дәлдік көрсеткіштеріне негізделген ең тиімді алгоритмдерді анықтауға бағытталған. Зерттеу мақсаттарына IDS өнімділігін жақсарту үшін алгоритмдерді ұсыну және енуді анықтау жүйесінің алгоритмін әзірлеу кіреді. Әдістеме әртүрлі DPI алгоритмдерін зерттеуді, деректер жиынын қалыпқа келтіруді және эксперименттер арқылы алгоритмдердің өнімділігін салыстыруды қамтиды. K-Nearest сияқты алгоритмдердің алдын ала нәтижелері алгоритмді әзірлеудегі үздіксіз ілгерілеушілікті көрсетеді.

Жоба зерттеу ортасының шектеулері сияқты қиындықтармен бетпе-бет келсе де, тақырыпты мұқият түсінуге деген ұмтылыс өзгермейді. Linux операциялық жүйелерін және мұқият зерттеулерді пайдалана отырып, зерттеу неғұрлым тиімді алгоритмдерді әзірлеуге әкелуі мүмкін шабуылдарды анықтау жүйелерінің тиімділігі туралы құнды ақпарат беруге бағытталған.

Аннотация

Быстрое распространение Интернета произвело революцию в современной жизни, однако оно также создает значительные риски из-за уязвимостей во все более сложном программном обеспечении и сетях. Устранение этих рисков требует надежных методов предотвращения атак. Брандмауэры служат важнейшими инструментами безопасности при защите сети, различая безопасные и менее безопасные сети и регулируя потоки информации. Однако сами по себе они не могут полностью защитить от вредоносных действий. Системы обнаружения вторжений (IDS), особенно те, которые включают в себя глубокую проверку пакетов (DPI), стали эффективным средством выявления и смягчения сетевых угроз.

Целью исследования является оценка эффективности алгоритмов DPI в обнаружении сетевых вторжений и реагировании на них. Анализируя существующую литературу и проводя эксперименты, исследование направлено на определение наиболее эффективных алгоритмов, основанных на показателях скорости и точности. Цели исследования включают предложение алгоритмов для повышения производительности IDS и разработку алгоритма системы обнаружения вторжений. Методика включает изучение различных алгоритмов DPI, нормализацию наборов данных и сравнение производительности алгоритмов посредством экспериментов. Предварительные результаты таких алгоритмов, как K-Nearest, демонстрируют постоянный прогресс в разработке алгоритмов. Несмотря на то, что проект сталкивается с такими проблемами, как ограничения исследовательской среды, стремление к тщательному пониманию предмета остается непоколебимым. Используя операционные системы Linux и тщательные исследования, исследование направлено на предоставление ценной информации об эффективности систем обнаружения вторжений, что потенциально приведет к разработке более эффективных алгоритмов.

Abbreviations

- **DPI** - Deep Packet Inspection - a network packet inspection method that includes packet payload information in the analysis.
- **TCP** - Transmission Control Protocol - connection-oriented transport layer protocol with reliability features.
- **UDP** - User Datagram Protocol - connectionless transport layer protocol.
- **HTTP** - HyperText Transfer Protocol - application-layer protocol designed for hypermedia information systems.
- **HTTPS** - HyperText Transfer Protocol Secure - a secure variant of HTTP protocol that provides confidentiality, integrity, authenticity guarantees.
- **HTTP/1.x** - in this paper this refers to first major HTTP version and its subversions HTTP/1.1 and HTTP/1.1.
- **HTTP/2** - HTTP protocol major version 2, published as IETF RFC in 2015, binary protocol unlike HTTP/1.x, and more feature-rich.
- **HTTP/3** - HTTP protocol major version 3, not yet finalized. Evolution of HTTP/2, uses QUIC protocol as transport.
- **FSM** - Finite State Machine - computational model of state machine with fixed number of states.
- **DFA** - Deterministic Finite State Automaton - A class of FSM where there is only one possible transition between states for given input.
- **NFA** - Non-deterministic Finite State Automaton - a class of FSM where multiple transitions to the next state are possible for given input.
- **DNS** - Domain Name System - a protocol for translating human-readable resource names into IP addresses.
- **TLS** - Transport Layer Security - a protocol for allowing secure communications. Commonly used with HTTP protocols.
- **DoH** - DNS over HTTPS - a protocol for performing DNS queries over HTTPS.
- **VPN** - Virtual Private Network - a technology for extending and connecting to a private network over internet.
- **ALPN** - Application Layer Protocol Negotiation - TLS extension that allows applications to negotiate which protocol they will use for communication.
- **packet** - in this paper the term refers both to TCP segments and UDP datagrams.

- **IPS** - Intrusion Prevention System - a security solution that monitors network and system activities to detect and prevent malicious or unwanted behavior in real-time.
- **IDS** - Intrusion Detection System - a device or software application designed to detect unauthorized access, breaches, or misuse of a network or system.
- **KDD '99** - Knowledge Discovery and Data Mining (Dataset)
- **U2R** - User to Root

Table of Contents

Declaration i

Acknowledgements ii

Dedication iii

Abstract iv

List of Abbreviations vii

1 Background and motivations 1
- 1.1 Introduction . . . 1
- 1.2 Research problems . . . 2

2 Related work 4

3 Packet Inspection 20
- 3.1 Packet Structure . . . 20
- 3.2 Packet Inspection Techniques . . . 23
- 3.3 Intrusion Detection system . . . 23
  - 3.3.1 Intrusion Detection methods . . . 24

4 Deep packet inspection 28
- 4.1 Overview of DPI . . . 28
- 4.2 Methods of Deep Packet Inspection . . . 28
- 4.3 Algorithms of DPI . . . 29
  - 4.3.1 Regular Expression Matching Algorithm . . . 29
  - 4.3.2 String Matching Algorithms . . . 29
  - 4.3.3 Finite State Machine . . . 30
  - 4.3.4 Machine Learning Algorithms . . . 30

5 Evaluation 32
- 5.1 Datasets . . . 32
- 5.2 Evaluation metrics . . . 33
- 5.3 Machine Learning Algorithms . . . 34

6 Results 36

6.1 Experiment Results on NSL-KDD Dataset 36

7 Conclusions and future work **38**
7.1 Conclusions 38
7.2 Future work 39

Bibliography **39**

Chapter 1

Background and motivations

1.1 Introduction

Cybersecurity involves organizing and assembling resources, processes, and structures to safeguard cyberspace and cyberspace-enabled systems from incidents where legal property rights do not match actual control or ownership [1].

Intrusion detection systems (IDS) analyze data and network behavior, playing a critical role in protecting networks from various cyber threats. As dependence on interconnected systems and digital communications continues to grow, the need for effective intrusion detection mechanisms has become increasingly important. It is anticipated that by 2022, internet connectivity will encompass 6 billion people, an increase from 5 billion in 2020, with projections suggesting that this number could rise to over 7.5 billion by 2030 [2]. With more devices connected and storing data, there is a heightened risk of security breaches and unauthorized access.
Traditional security measures such as firewalls, access controls, behavioral analytics, and antivirus software are commonly employed within the computing industry. Firewalls and spam filters primarily use basic rule-based algorithms to regulate traffic by filtering protocols, ports, and IP addresses. However, their effectiveness remains a question as cyber threats evolve. The proliferation of sophisticated malware, Man-in-the-Middle, and DoS attacks continue to advance, posing significant risks to business infrastructures. Additionally, the prevalence of zero-day attacks targeting internet users is on the rise, making not just small businesses but also major banks and specific entities increasingly vulnerable.
According to the "2022 Official Cybercrime Report" by Cybersecurity Ventures, the financial repercussions of cybercrime are expected to surge by 15% annually over the next five years, potentially reaching $10.5 trillion annually by 2025, a significant rise from $3 trillion in 2015. By 2023, cybercrime's global cost is projected to hit $8 trillion, marking the largest economic wealth transfer in history, which threatens the drive for innovation and investment, surpassing the yearly damages from natural disasters [2].
Moody's reports that certain sectors are particularly susceptible to cyber threats. Critical infrastructure sectors, such as utilities and hospitals, face severe vulnera-

bilities. Likewise, the banking, telecommunications, technology, chemicals, energy, and transportation sectors are significantly exposed to cyber risks [3].

Deep packet inspection (DPI) is a technology that allows detailed inspection of the contents of network packets as they pass through inspection checkpoints. The goal of DPI is to go beyond basic packet header inspection, allowing deeper insight into the payload to identify, classify, or block specific types of traffic based on its content, in real time. This capability is critical for network security, management, and policy compliance [4]. Deep packet inspection (DPI) techniques offer a comprehensive approach to network traffic analysis by examining both packet headers and payloads. These techniques enable the inspection of content such as HTML for malicious scripts or scanning file transfers for malware, regardless of the expected port. DPI transcends individual packet data, utilizing accumulated information to make informed decisions about network traffic management. This analysis extends to the entirety of network communications, allowing for sophisticated traffic shaping and policy enforcement. Some common applications of DPI include:

- Protocol Analysis: DPI can scrutinize known protocols for deviations from expected behaviors, signaling possible attacks or anomalies in network traffic.
- Network Analysis: DPI facilitates in-depth analysis of network traffic patterns and behaviors, providing insights into network performance, usage trends, and potential security risks.
- Traffic Throttling: DPI empowers network administrators to throttle network speeds for specific content types, enabling the prioritization of critical traffic and the mitigation of bandwidth-intensive activities. [4]
- Intrusion Detection Systems (IDS): DPI enables IDS to monitor network traffic for malicious payloads, allowing for the detection and prevention of potential security breaches.

1.2 Research problems

Integrating deep packet inspection (DPI) into intrusion detection systems (IDS) poses several research challenges. These include optimizing performance to process large volumes of network traffic in real-time while maintaining high detection accuracy, accurately identifying a wide range of protocols and applications, including encrypted traffic, developing techniques to resist adversarial evasion attempts, balancing the need for threat detection with user privacy rights, scaling DPI architectures to handle high-speed networks efficiently, integrating DPI with other detection techniques for comprehensive threat detection, updating threat models dynamically to detect new and emerging threats, enhancing the interpretability of DPI-based IDS outputs for security analysts, and addressing real-world deployment challenges such as network heterogeneity and operational constraints. Tackling these challenges requires interdisciplinary collaboration and advancements in algorithm design, machine learning, and network security research.

This thesis focuses on the intrusion detection system which is use deep packet inspection algorithms, structured into the subsequent chapters. In Chapter 2, we

describe the relevant work of authors who have studied a topic related to ours, and also show this comparison in graphical form, explaining the main concepts of the result of the work. Chapter 3 describes the package inspection framework and methods. Chapter 4 describes deep packet inspection (DPI) techniques and algorithms. It also shows popular DPI systems and the difference between them. Section 5 of the chapter discusses the evaluation and data set we chose for experimentation and comparison, and shows the implementation process. After we described the results of the experiment based on metrics in Chapter 6. The final chapter, presents conclusions and future work on deep packet inspection and prioritizes future research.

Chapter 2

Related work

Intrusion Detection Systems (IDS) play a vital role in safeguarding network infrastructure from malicious activities. With the advancement of technology, IDS incorporating Deep Packet Inspection (DPI) have emerged as a promising approach for detecting sophisticated attacks. Many research studies utilize a mix of different attack detection techniques to achieve high accuracy in identifying attacks. This indicates that intrusion detection methods often overlap with each other. This section provides a review of relevant literature focusing on intrusion detection systems with DPI, culminating in a comprehensive understanding of the field.

Laghrissi et al. [5] proposed an efficient algorithm for intrusion detection systems using attention mechanism.The authors tried to show that network attacks are illegal actions against digital assets within an organizational structure, with the explicit goal of hacking systems. Cyber attacks, whether carried out by individuals, communities, states or clandestine organizations, serve to manipulate, damage or steal private data. They also suggested that combating these threats with the help of Intrusion Detection Systems (IDS) are becoming essential and highly effective mechanisms. In their article They noted that operating as both software applications and hardware entities, IDS vigilantly monitors network traffic to detect malicious activity or violations of established policies. Moreover, these systems adapt to different deployment scenarios, appearing as either host or network objects. Host-based intrusion detection systems are built into client machines, while network-based counterparts are located within the network architecture itself. Also in their article, the researchers say that despite the effectiveness demonstrated in recent years by IDS based on deep learning methodologies, notable problems remain. These methodologies often exhibit a noticeable tendency to generate significant false negatives, thereby reducing the overall effectiveness and reliability of network security protocols. To address these limitations, this paper proposes a novel detection system based on the integration of long short-term memory (LSTM) architecture and attention mechanisms. In addition, the authors conducted a comprehensive evaluation using four dimensionality reduction algorithms, namely chi-square, uniform manifold approximation and projec-

tion (UMAP), principal component analysis (PCA), and mutual information. The proposed methodologies are thoroughly evaluated using the NSL-KDD dataset. Their results demonstrate compelling performance metrics, highlighting the effectiveness of the proposed detection model. Of particular note is the exceptional performance observed when using attention mechanisms, see Figure 2.1 in combination with all available features, as well as when using PCA, which includes three main components. Notably, this configuration achieves 99.09% accuracy for binary classification tasks, with comparable accuracy of 98.49% achieved for multiclass classification. These results highlight the significant potential of the proposed framework in hardening network security infrastructure against a wide range of adversarial attacks.

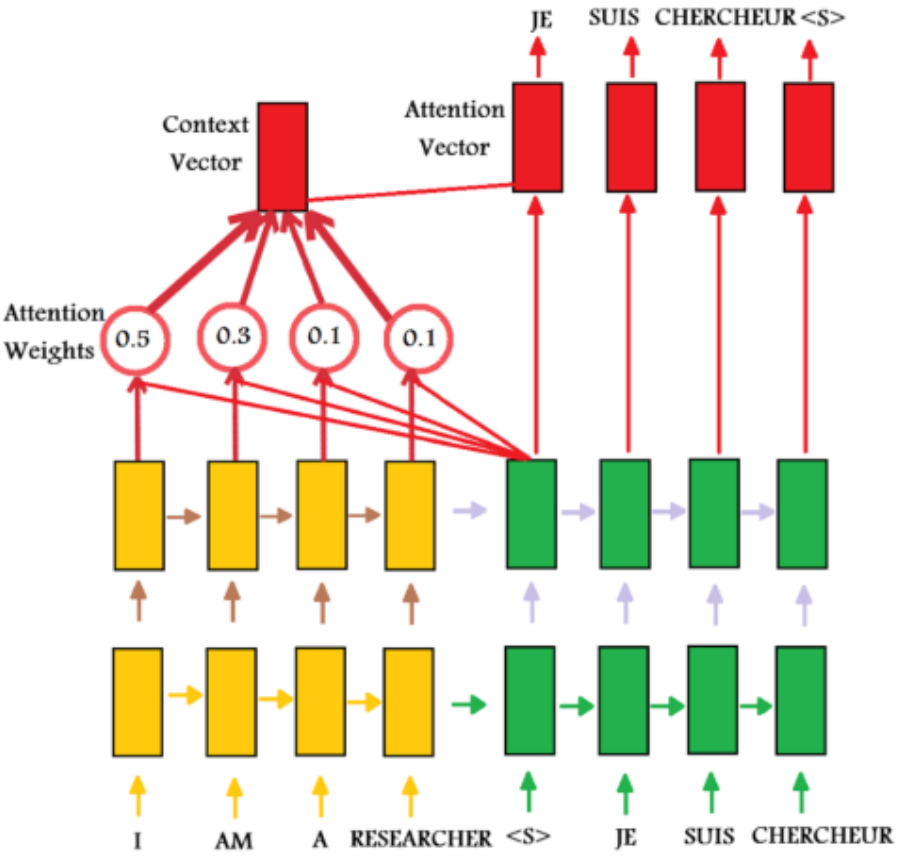

Figure 2.1: Attention mechanism architecture

In [6] takes a comprehensive look at intrusion detection technologies, methodologies, and approaches, and examines new attack vectors, defenses, and recent advances in the field. In addition, the article carefully examines the available data sets, talks about known intrusion detection system (IDS) tools, and also carefully examines, in the opinion of the authors, the advantages and disadvantages associated with specific IDS implementations. The authors assert that this scientific review serves as a valuable roadmap for both researchers and industry professionals specializing in the field of IDS, providing invaluable guidance and direction for future endeavors and initiatives.

The authors of [7] explained that deep packet inspection (DPI) is a critical component of content-aware network applications such as copyright protection, intrusion detection systems (IDS), and various other applications. According to the authors, DPI works by inspecting the payload and IP headers of network packets, comparing them with known signatures to identify potentially malicious packets. If a packet matches a signature in the attack database, it is flagged and processed accordingly, either discarded or allowed to pass through the network flow. El-Maghraby detailed a process that delves into the contents of Layer 4 of an IP packet, analyzing source and destination ports, addresses, and protocol types to classify application types based on port numbers. The article shows how signature comparison methods are mainly used in DPI, and the authors of the article also found that the most widely used method is regular expressions. These algorithms scan packet data for patterns that match known threats. This process is computationally intensive and critical to maintaining network security. This article describes several algorithms that are used for signature, each of which has its own set of advantages and limitations. The article outlines key design goals such as efficiency, scalability, and the ability to handle encrypted traffic. It also provides an overview of various matching algorithms, highlighting their strengths and weaknesses. This survey serves as a foundational guide to understanding the complexities and essential functions of DPI in securing today's network environments, while in this article [8], the authors pay more attention to unsupervised methods. As this article examines the growing frequency and variety of network attacks over the past five years, criticizes the robustness of existing IDSs that rely on legacy attack classes, and therefore provides a comprehensive overview of unsupervised and hybrid intrusion detection techniques, emphasizing the importance of feature development and the need to evolve IDSs toward correlation and attribution using advanced data analytics, and proposing three new attack classes for outbound network communications.

While [9] explores the integral role of deep packet inspection (DPI) intrusion detection and prevention, the authors have done an excellent job of highlighting the various signature detection techniques and algorithms, each with their own advantages and limitations in terms of time, accuracy, and space requirements. As technology advances, so do these methods, critically adapting to the expanding web services landscape. In the paper, Thaksen J. Parvat and Pravin Chandra highlight the challenges of managing secure networks today, emphasizing that goals vary depending on infrastructure management and security policies. DPI plays a critical role in identifying payload traffic, ensuring network security, privacy and quality of service (QoS) through features such as protocol discovery, antivirus, anti-malware and IDS/IPS. It discusses how detection engines use both signatures and heuristics, noting that most algorithms require separate training and testing steps, effectively doubling the time required. Finally, The authors of the paper propose their new model aimed at improving DPI performance in intrusion detection systems, eliminating current limitations and improving overall network security. This work differs from others in that it is determined not only by the theoretical part but also by the practical one, and their results obtained through experiment show the best result in accuracy and in terms of time complexity, ac-

cording to the authors themselves, they improved these results in the Figure 2.2, which comparing their algorithm with others.

Classifier	Correctly Classified Insistence	%	Incorrectly Classified Insistence	%	Time in ms
BayesNet	24326	96.5624	866	3.4378	0.63
NaiveByes	22570	89.5919	2622	2.8144	0.2
Decision Table	24925	98.9004	257	1.0202	11.55
Tree J48	25081	99.615	111	0.4406	1.55
Our Algorithm	24815	98.5035	377	1.4965	0.18

Figure 2.2: Comparative Classification Results and Time

If previous research conducted experiments and surveys on conventional algorithms, then this article [10] shows that even the most secure algorithms cannot cope with growing security problems, so it is worth resorting to methods such as machine learning (ML) algorithms. This paper demonstrates the use of IDS in applications such as fog computing, Internet of Things (IoT), big data, smart cities and 5G networks, in particular evaluating the effectiveness of algorithms such as linear discriminant analysis (LDA), classification and regression trees (CART) and Random Forest using the KDD-CUP dataset. The authors analyzed the articles and identified which machine learning algorithm was the most accurate, see Figure 2.3.

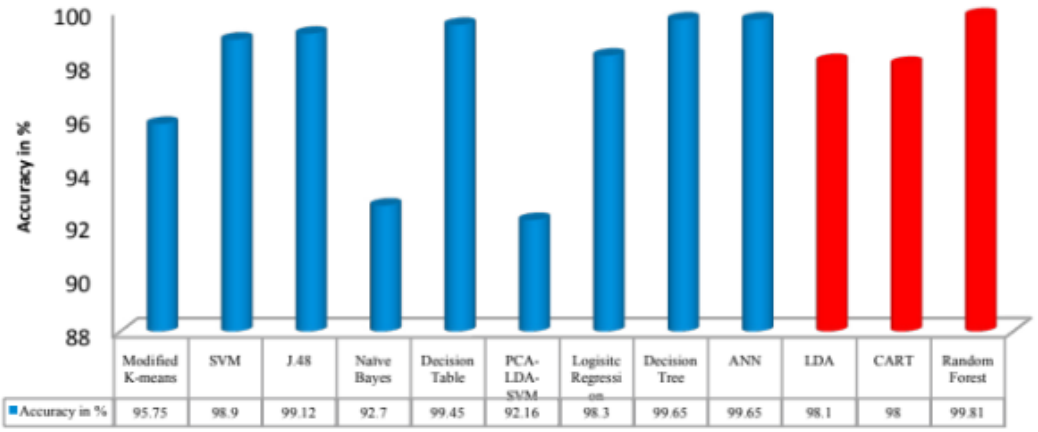

Figure 2.3: Performane Comparisons of various ML algorithms in IDS

Likewise Md. Rayhan Ahmed, et al. [11] highlights the transformative impact of software-defined networking (SDN), a paradigm that decouples the control plane from the data plane, enabling programmable and manageable networks. However, the centralization of the SDN control plane, which coordinates network functions through APIs, also makes it a prime target for security attacks. The article provides a comprehensive overview of significant research conducted from 2015 to 2021, focusing on the application of traditional machine learning (ML) and deep

learning (DL) methods to develop IDS solutions for SDN security. It includes two detailed taxonomies of IDS solutions and ML-DL methods, classifying them based on learning methods. The survey also examines key benchmark sets used in building IDS for SDN and discusses current issues and challenges associated with IDS in the context of SDN security.

This research paper [12] examines the problem of detecting polymorphic malware that constantly changes its signature to bypass traditional detection methods. For malware data, it is not enough to use conventional intrusion protection methods, so this article explores the use of various machine learning methods to identify malicious threats, and the algorithm with the highest detection accuracy is selected for implementation. The work uses a confusion matrix to evaluate false positives and false negatives, which provides insight into the performance of the system. The results show that the Decision Tree (DT), Convolutional Neural Network (CNN), and Support Vector Machine (SVM) algorithms achieved high detection accuracy rates of 99%, 98.76%, and 96.41%, respectively, as well as low false positive rates. demonstrating their effectiveness in enhancing network security by detecting sophisticated malware.

The authors of [13] focus on the application of data mining techniques for intrusion detection. The research provides a thorough analysis of the advantages and disadvantages of various data mining techniques used in this context. The basis of the paper is the presentation of a new approach that uses data mining to improve the performance of intrusion detection systems (IDS).

In the [14] , the authors highlight the role of machine learning (ML) and deep learning (DL) in improving IDS performance. This paper explains the concept of IDS and classifies various ML and DL techniques used in the development of Network IDS (NIDS). It provides a comprehensive overview of recent research related to NIDS, discussing the strengths and limitations of each proposed solution. In addition, the paper reviews recent trends and advances in ML and DL based NIDS, focusing on methodologies, evaluation metrics, and selection of datasets. Finally, Ahmad Zeeshan, et al., highlight current research challenges and suggest future directions for improving ML and DL based NIDS.

Moreover, [15] examines the vulnerability of deep packet inspection (DPI) mechanisms built into security appliances to denial of service (DoS) attacks of algorithmic complexity, highlighting a critical security weakness in enterprise and cloud networks. It introduces a system and multi-core architecture designed to protect against such sophisticated attacks. The paper details the integration of this defense system with two different DPI mechanisms and shows how a simple low-bandwidth cache miss attack can defeat the Aho-Corasick (AC) pattern matching algorithm that is fundamental to many DPI mechanisms. To mitigate such attacks, the authors develop a compressed version of the AC algorithm, which improves performance under attack but has worse performance under normal traffic. To address this issue, they propose MCA^2 (Multi-Core Architecture to Mitigate Complexity Attacks), which dynamically combines the classic AC algorithm with a compressed variant to provide a fault-tolerant solution. The paper shows that

this architecture significantly improves efficiency (up to 73%) against cache miss attacks and can be generalized to effectively counter a variety of algorithm complexity attacks. The algorithm research in the current work of the Aho-Corasick algorithm is based on the seminal work of researchers Aho, Alfred V. and Corasick, Margaret J. [16]. Their original research, detailed in their seminal paper, presented an efficient method for finding all occurrences of any finite set of keywords in a text string. This method involves building a finite-state pattern matching machine from keywords and then using that machine to process a text string in a single pass. Building a pattern matching machine takes time proportional to the sum of the keyword lengths. It is important to note that the number of state transitions made by the pattern matching engine when processing a text string is independent of the number of keywords, which greatly improves efficiency. This algorithm, in particular, has been used to improve the speed of library retrieval programs by 5 to 10 times, demonstrating its practical application and impact. Thus, this scientific discovery gave rise to the study of this algorithm. Building on this foundational work, other researchers have conducted surveys and overview articles that explore related areas, such as the field of intrusion detection systems (IDS) and network security. One such study investigates the crucial role that deep packet inspection (DPI) techniques play in protecting computer networks from a variety of online threats. This research [17] highlights the criminal undertones of intentional network intrusions, which can aim to take over systems, obtain sensitive data, disrupt network functions, or incapacitate websites and servers.
The study emphasizes the growing frequency of attacks on IT infrastructure and underscores the critical need for effective intrusion detection. It identifies a gap in the existing body of knowledge regarding the effectiveness of DPI algorithms in countering vulnerabilities. The analysis presents a grim picture of current network security, including alarming statistics on ransomware attacks, cyberattacks targeting small organizations, and data breaches leading to substantial financial losses. It also notes that human error remains a significant weakness in network security. Intrusion detection systems (IDS) are highlighted as essential tools for preventing unauthorized access and safeguarding systems. IDS involves the identification and evaluation of system or network events to detect potential intrusions and stop malicious activities. The study underscores the constant evolution of attack strategies by malicious actors, making network security a major concern. The main goal of this research is to investigate and compile the body of knowledge on various DPI intrusion detection techniques. It proposes the creation of a morphological framework for IDS to deepen our understanding of these systems. The study provides a comprehensive explanation of different DPI types and their corresponding algorithms, such as Aho-Corasick, Regular Expression Matching, Wu-Manber, p3FSM, and BOM.

The classification of intrusion detection systems is very important, it is presented in the most understandable form in [18] , as a result of the described work we can conclude that for an attack detection tool which need to know attacks well. Using the example in Figure 2.4, can be seen how the authors of this article divided the detection system. It is also important to always update the types of new attacks. A complete guide allows researchers to distribute not only attacks but

also detection systems and the differences between methods. It also reviews the available datasets and known IDS tools and outlines their advantages and disadvantages. This article provides valuable information and a roadmap for researchers and industry professionals involved in IDS.

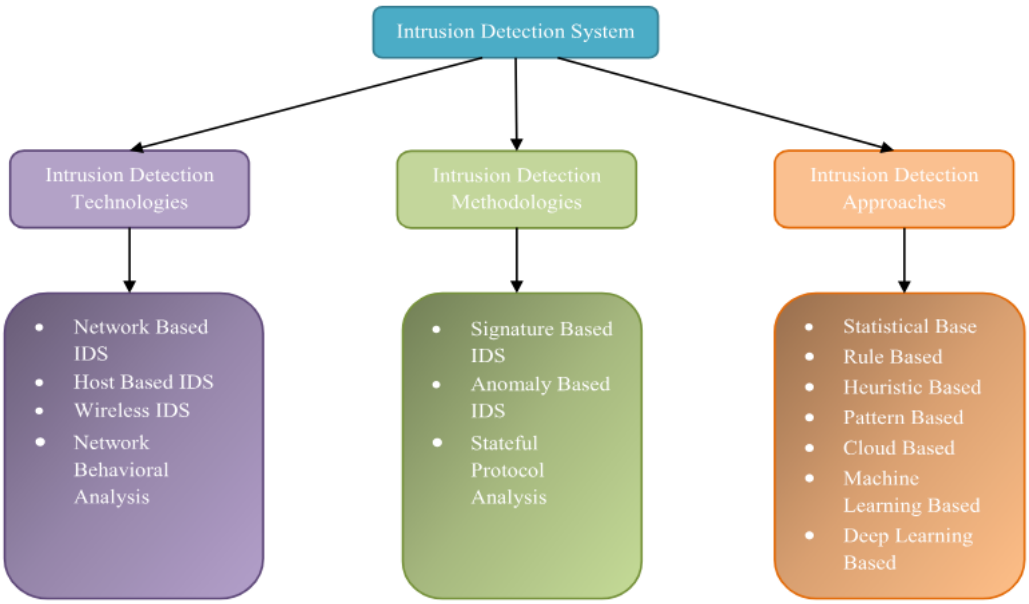

Figure 2.4: Classification of intrusion detection system

Regular expressions are commonly used to describe signatures due to their powerful and flexible nature, but this flexibility poses significant challenges to efficient implementation. Despite numerous empirical proposals, achieving wireline speed matching for large-scale regular expressions remains a challenge, especially as network connection speeds and pattern scales continue to increase, widening the gap between matching throughput and connection speed. This [19] provides comprehensive information on the application and technical aspects of DPI and regular expression matching, offering readers a holistic understanding of the field. The authors delve into the problems posed by state explosion of state machines used in regular expression matching, classifying state-of-the-art solutions into methods for mitigating state expansion and preventing state explosion. In addition, the paper discusses proposals for using parallel platforms such as field programmable gate arrays (FPGAs), GPUs, general purpose multiprocessors, and ternary content addressable memory (TCAM) to speed up the mapping process. The survey also provides practical recommendations for effectively deploying these platforms, making it a valuable resource for researchers and practitioners looking to improve DPI performance in the face of increasing network demands.
Various DPI applications are driven by different stakeholders in Figure 2.5, illustrated by solid, dotted, and red lines. The solid line indicates the primary entities promoting these applications, such as businesses and government agencies. The dotted line shows secondary supporters who benefit indirectly, like ISPs or third-

party security firms. The red line highlights illegal activities, indicating areas where DPI is used for surveillance or to counteract malicious actions. This visual representation helps clarify the diverse roles and interests in DPI deployment and usage.

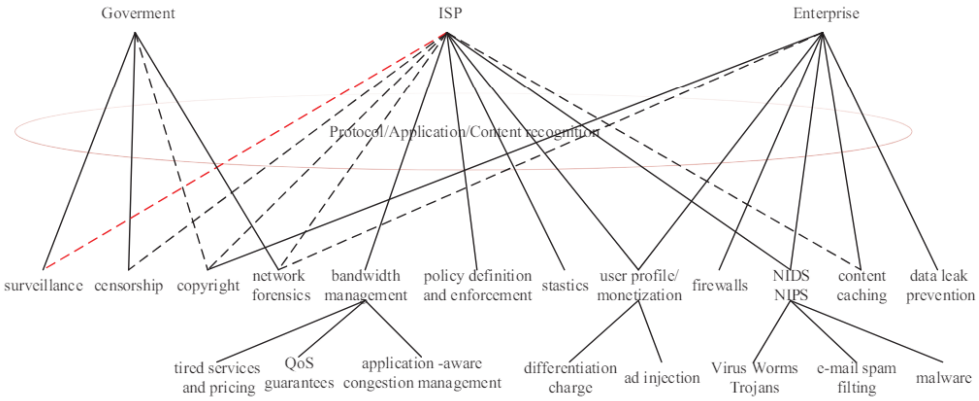

Figure 2.5: Various DPI applications from different perspectives, solid line represents primary promoting party, dotted line means secondary promoting party, and red line means illegal activity

Unlike other articles, this examines the weakness in DPI systems when subjected to low bandwidth cache-miss attacks, which target the Aho-Corasick (AC) pattern matching algorithm—a cornerstone of many DPI engines. To mitigate these vulnerabilities, the paper presents a compressed variant of the AC algorithm that improves worst-case performance under attack conditions, although it performs worse under normal traffic.
To demonstrate these issues, examples from Becchi's dissertation are employed, showcasing Non-deterministic Finite Automaton (NFA), Figure 2.6 and Deterministic Finite Automaton (DFA), Figure 2.7, representations of regular expressions like `abcd+`, `ab[c-z]e+`, `bc+da`, and `bc+`. Figures in the paper visually depict these automata, with states and transitions clearly marked. NFAs have multiple potential active states, which can complicate memory access and increase processing time, especially in worst-case scenarios where all states might be active simultaneously.

To investigate the potential benefits of using SDN as security in industrial networks, the authors of this article [20] conducted an experiment in which they provided how they developed this system, and they also provided the results of the experiment conducted on a large-scale physical implementation. This article points to the urgent need for cybersecurity measures in ICS environments, highlighting the shortcomings of traditional approaches to IT security. A significant contribution of this work is its detailed discussion of SDN integration in industrial networks, the implementation plan can be seen in the Figure 2.8, highlighting benefits such as improved network visibility, improved control over network traffic, and dynamic security policy implementation. The authors of the pilot test

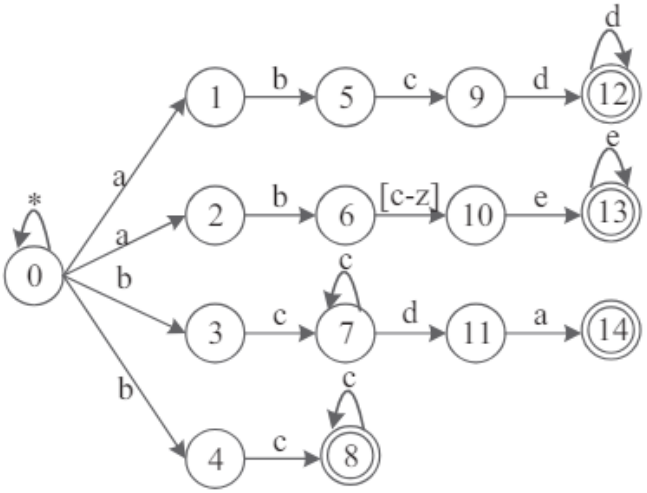

Figure 2.6: A NFA accepting regular expressions of abcd+, ab[c-z]e+, bc+da, and bc+

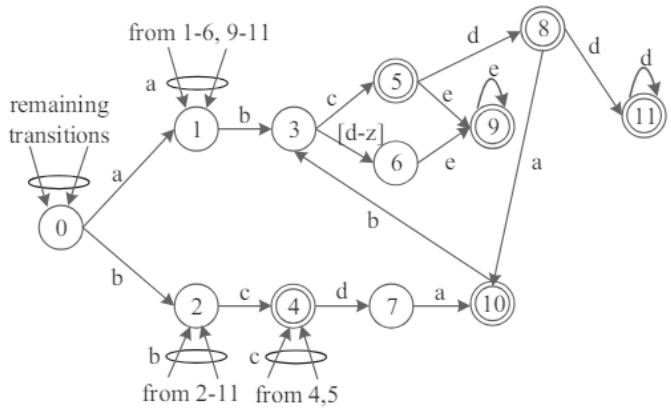

Figure 2.7: A DFA accepting regular expressions of abcd+, ab[c-z]e+, bc+da, and bc+

the viability of SDN-based IDS in an ICS environment by detecting ICMP Flood attacks and packet payload changes using signature matching techniques. The results indicate the potential viability of SDN-based IDS technology for ICS intrusion detection and also highlight the need for further research into the scalability of the architecture. This article did not provide a full assessment of the various attack vectors and the broader range of experimental scenarios. Additionally, a deeper understanding of integration issues between SDN and existing ICS infrastructure, including compatibility issues, deployment costs, and impact on system performance, will enhance the analysis. Overall, Markel Sainz's paper presents a compelling case for implementing SDN technology to improve ICS security and provides a strong foundation for further research, demonstrating potential benefits and identifying key areas for improvement. The practical findings from the pilot are valuable for academic researchers and industry professionals seeking to develop and deploy advanced IDS solutions in ICS environments.

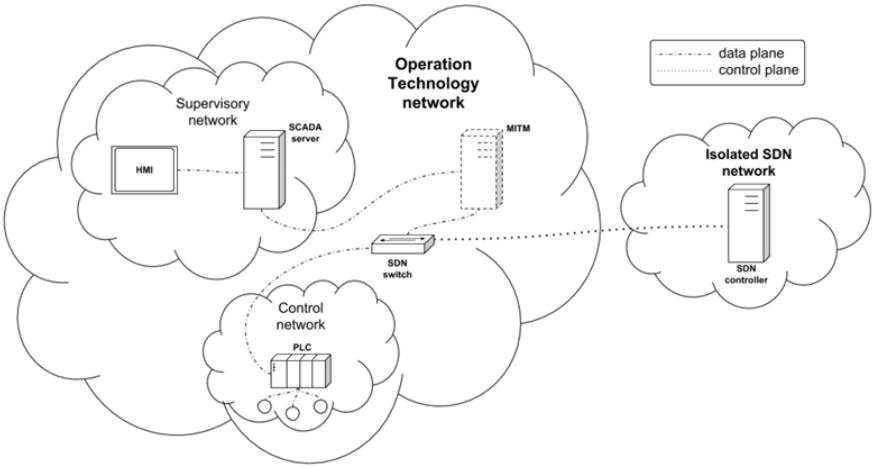

Figure 2.8: Implemented topology

Machine learning and deep learning algorithms can significantly improve the accuracy and efficiency of IDS. Traditional IDS rely heavily on predefined signatures and rule-based systems to detect known threats. However, these systems often struggle to identify new or evolving threats. ML and DL models, particularly those employing supervised, unsupervised, and reinforcement learning techniques, can learn from vast amounts of data and recognize patterns indicative of both known and unknown attacks. Based on this, the researchers in the [21] focused on developing an IDS that uses both deep learning and traditional machine learning approaches to effectively detect network intrusions. Key contributions include the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks for their ability to capture spatial hierarchies, temporal dependencies, and long-term dependencies in network traffic data. , respectively. These deep learning methods are complemented by traditional machine learning algorithms such as support vector machine (SVM), random forest and XGBoost. SVM is used for its robustness in classification problems, Random Forest for its ensemble learning capabilities, and XGBoost for its efficiency and accuracy in processing large data sets. Also, the IDS developed by the authors was evaluated using the NSL-KDD dataset, a benchmark dataset widely used for network intrusion detection research. The study reports achieving near-state-of-the-art performance in detecting network attacks, demonstrating the effectiveness of combining deep learning and traditional machine learning methods.
The article "Intrusion Detection System for Wireless Networks" examines the enhancement of network security through intrusion detection systems (IDS) using both traditional machine learning algorithms and advanced deep learning techniques. The research tested models like CNN, RNN, LSTM, SVM, Random Forest, and XGBoost on the NSL-KDD dataset to identify network attacks.

Results of their experiment:

- CNN-1: Training accuracy of 99.85%, validation accuracy of 99.68%, and test accuracy of 81.41%.

- CNN-RNN-1: Training accuracy of 98.97%, validation accuracy of 99.67%, and test accuracy of 79.23%.
- CNN-BN-1: Test accuracy of 78.97%, with training and validation accuracies of 99.87% and 99.68%, respectively.
- NOG-LSTM: Highest test accuracy at 85.68

Experiment Details:

- Tested NSL-KDD dataset in formats of 11x11 and 121 features.
- Used Adam optimizer, batch size of 1000, and a learning rate of 0.001 over 100 epochs, except for NOG-LSTM (batch size 512 over 5 epochs).
- Batch normalization (BN) was applied in some models to evaluate its effect.

Deep learning models, particularly the NOG-LSTM, significantly enhanced detection accuracy of network attacks. Combining convolutional and recurrent neural networks proved effective. The study underscores the importance of machine learning and deep learning in developing robust IDS for wireless networks, achieving near state-of-the-art results in network security. Another research work [22] where addresses the challenge of enhancing the efficiency and accuracy of intrusion detection systems (IDS) for both network and host environments. For network intrusion detection, it introduces an integrated deep intrusion detection model based on Stacked Denoising Autoencoder and Extreme Learning Machine (SDAE-ELM) to mitigate the long training times and low classification accuracies of traditional deep neural networks, enabling quicker response to intrusions. For host intrusion detection, the model uses Deep Belief Network and Softmax (DBN-Softmax) to boost detection accuracy. To enhance the training efficiency and performance of these models, a small batch gradient descent method is employed for network training and optimization. The models are tested on various datasets, including KDD Cup99, NSL-KDD, UNSW-NB15, CIDDS-001, and ADFA-LD, demonstrating superior performance over traditional machine learning models.

Results:

- Network Intrusion Detection: The SDAE-ELM model significantly reduces training time and increases classification accuracy.
- Host Intrusion Detection: The DBN-Softmax model effectively improves detection accuracy.
- Performance: Both integrated models outperform classic machine learning models across multiple benchmark datasets.

The research indicates that the proposed models provide a robust and efficient solution for both network and host intrusion detection, ensuring timely and accurate responses to intrusion activities. The integrated deep intrusion detection models introduced in this paper are structured into three main modules, and the structure is shown in Figure 2.9.

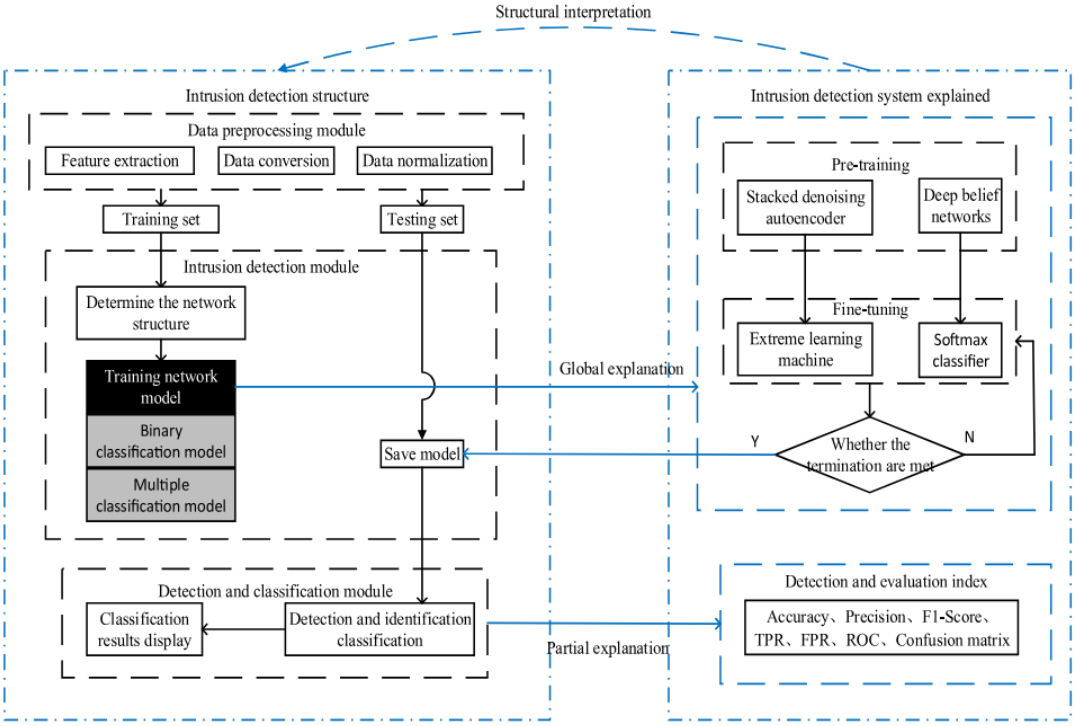

Figure 2.9: Intrusion detection model structure diagram

Using deep neural networks (DNNs), IDS can flexibly learn and identify both known network behavior and zero-day network behavior, effectively reducing the risk of system compromise. The UNSW-NB15 dataset, which captures state-of-the-art network communications behavior with synthetically generated attacks, is used to demonstrate the effectiveness of the model, the results demonstrated here 2.1.

Dataset	**Accuracy (%)**	**Optimizer**	**Learning Rate**	**Dropout Rate**
Pre-Partitioned UNSW-NB15	94.4	Adam	0.001	Varying
User-Defined UNSW-NB15	95.6	Nadam	0.005	Similar Pattern

Table 2.1: Model Performance on UNSW-NB15 Datasets

In this work [23], the authors propose to use deep learning architectures to develop an adaptive and robust network intrusion detection system (IDS) to detect and classify network attacks. Detailed accuracy of learning curves and detection rates for various models and datasets.

Model Testing for Pre-Partitioned UNSW-NB15 Datasets

- Benchmark Performance: Deep learning models on the UNSW-NB15 dataset show accuracies ranging from 78.4% to 98.47% ,.
- Proposed Model Performance: Achieved 94.4% accuracy on the testing dataset, shown in Figure 2.10.

- Algorithm Details: Utilized 'categorical cross entropy' with the Adam optimizer, a learning rate of 0.001, and a varying dropout rate to prevent overfitting.
- Performance Observations: The model's multiclass classification accuracy and the detection rates for different attack types showed lower detection rates for underrepresented classes, shown in Figure 2.11 .

Model Testing for User-Defined UNSW-NB15 Datasets

- Benchmark Performance: Models on the user-defined partition showed higher performance than pre-partitioned data.
- Proposed Model Performance: Achieved 95.6% accuracy on the testing dataset, shown in Figure 2.12.
- Algorithm Details: Used 'categorical cross entropy' with the Nadam optimizer, a learning rate of 0.005, and a similar dropout rate pattern as the pre-partitioned model.
- Performance Observations: Similar to the pre-partitioned dataset, the detection rate for underrepresented classes was significantly lower, shown in Figure 2.13.

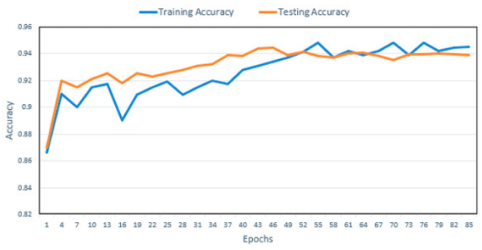

Figure 2.10: Learning curve accuracy for the multiclass classification model (pre-partitioned datasets)

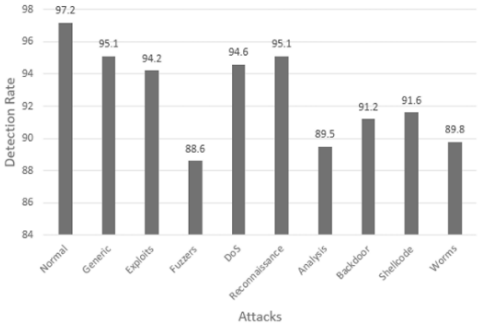

Figure 2.11: Detection rates for the types of attacks (pre-partitioned datasets)

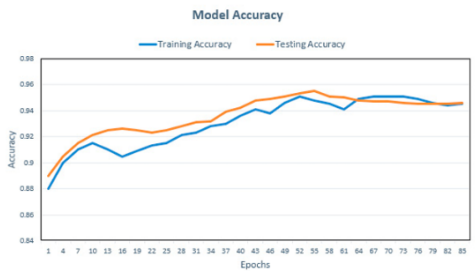

Figure 2.12: Learning curve accuracy for the multiclass classification model (user-defined datasets)

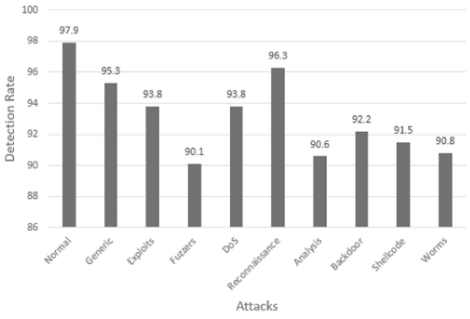

Figure 2.13: Detection rates for the types of attacks (user-defined datasets)

This article [24] explores the use of firewall clusters as a means to improve cybersecurity and protect against network attacks. It highlights that deep packet inspection (DPI) is a powerful but performance-intensive technique used in firewalls. The study is notable because it is one of the first to use real network equipment under realistic conditions to compare firewall clusters with single firewalls using DPI. The study measured the impact of latency under various traffic conditions and found that DPI and firewall clusters could significantly increase latency and reduce speed by more than 10% in some scenarios. These results are valuable for network engineers and researchers in designing efficient and secure networks.

The significance of wireless networks has rapidly expanded, making them vital to modern life. However, this proliferation presents substantial security challenges. Intrusion Detection Systems (IDS) are crucial for detecting and preventing threats in wireless networks, yet they face challenges due to infrastructure and computational limitations. This literature review [25] aims to provide a comprehensive analysis of IDS in wireless networks, discussing defense mechanisms, intruder types, intrusion behaviors, and various IDS frameworks. It also explores effective IDS design, state-of-the-art techniques, critical attacks, and the limitations of modern wireless networks, encouraging further research in cybersecurity.

During our research and experiments, we search information in books, one of which [26] is devoted to data mining and its application in intrusion detection systems (IDS). The book covers such fundamental topics as data cleaning, integration, selection, transformation and extraction. With an emphasis on machine learning and pattern recognition techniques, the book addresses the shortcomings of existing IDS methods, proposing new approaches using reinforcement learning and coarse-grained sets. This resource is intended for graduate and doctoral students in technical specialties, as well as network security specialists. The five chapters cover the basics of intrusion detection, data preprocessing, pruning techniques, Q-learning classifiers, and future research directions with an emphasis on improving the accuracy and efficiency of IDS, which helped in writing the dissertation as well as in gaining a deeper understanding of the topic.

Intrusion Detection Systems (IDS) are essential for defending against cyber attacks, which constantly evolve. This paper [27] evaluates the effectiveness of various machine learning classifiers using the KDD intrusion dataset to improve IDS. The experiments focused on performance metrics such as false negatives and false positives to enhance detection rates. The results showed that the decision table classifier achieved the lowest false negative rate, while the random forest classifier attained the highest average accuracy. The findings highlight the importance of selecting appropriate classifiers for effective IDS implementation.

According to the authors [28], the use of machine learning methodologies for anomaly detection has gained popularity in recent years. In this article, the authors conduct a thorough review of research based on intrusion detection systems using machine learning algorithms and present a comparison depending on the dataset used, data reduction approaches, the type of classifiers used, and the results achieved by such different algorithms.

According to the authors of [29], intrusion detection systems are critical to network security by helping to identify normal and malicious network traffic. The authors of the research paper explore machine learning algorithms—logistic regression, Gaussian Naive Bayes, support vector machine, and random forest—to build predictive intrusion detection models using the NSL-KDD dataset. The research experiments show that the Random Forest classifier outperforms others in accurately classifying network traffic as normal or attacker, highlighting its effectiveness in improving intrusion detection systems.

Kumar, et.al [30], in their research, focus on enhancing network security through an Intrusion Detection System (IDS) that identifies normal and attack data packets. Using the KDD-99 dataset, they conducted experiments with a sample size of 32,640, equally divided into training and testing sets. They applied SVM and kNN classifiers with PCA for dimensionality reduction. Their results showed that the PCA-kNN combination achieved the highest accuracy of 90.07% using cosine distance. This study demonstrates the effectiveness of machine learning techniques in distinguishing between normal and malicious network traffic.

This article [31] discusses the integration of Network Intrusion Detection Systems (NIDS) and firewalls to create an Intrusion Prevention System (IPS), which enhances network security by performing deep packet inspection. While effective, modern IPS solutions are costly due to the need for advanced hardware. The authors propose an affordable alternative by optimizing the open-source NIDS, Snort, with a Linux firewall using IPTABLES. Their integrated system, tested with DARPA99 datasets, demonstrated satisfactory results, offering a cost-effective solution for medium and small-sized companies.

According to the research work [32], securing network infrastructure is essential in the face of prevalent cyberattacks such as denial of service (DoS), backdoors, and buffer overflow attacks. The text outlines the complete process for detecting and mitigating these threats using automatic detection techniques. By integrating intrusion prevention systems (IPS) within firewalls, it aims to enhance network security by analyzing packet flows to detect and preemptively block malicious traffic. The study evaluates the effectiveness of different machine learning algorithms in IPS, presenting their error and accuracy rates in detecting attacks.

With the influx of false positives, the task of security analysts becomes more difficult, as they must analyze multiple alerts to identify genuine threats and take appropriate action. Thus, the authors [33] provided a two-phase automatic alert classification system designed to help analysts more effectively identify false positives. In the first stage, alerts collected from one or more sensors are normalized and similar alerts are grouped into meta-alerts. These meta alerts are then passively checked against the asset database to filter out irrelevant alerts. Additionally, an additional step of alert summarization is performed to facilitate root cause analysis, further reducing the number of false positives for some human interaction.
After the first stage, the scientists moved to the second stage, in which a reduced set of alerts is labeled and passed to an alert classifier, which uses machine learning

techniques to create classification rules. This automated classification helps analysts significantly streamline the alert assessment process. The system has been field tested and proven effective in significantly reducing both the total number of alerts and the number of false positives, thereby easing the workload of analysts.

The research [34] in this paper shows that content addressing can effectively create automata that are both very compact and capable of achieving high throughput. In particular, the study shows that for applications involving thousands of patterns defined by regular expressions, CD2FAs require only 10% of the memory required by conventional compressed DFAs, while still delivering the same performance as uncompressed DFAs. The paper presents content-deferred DFA (CD2FA), which offers a compact representation of regular expressions while maintaining the high throughput of traditional uncompressed DFAs. CD2FA uses the contents of sequential states in D2FA for addressing rather than relying on a common identifier. This approach makes selected information available earlier in the state traversal process, avoiding unnecessary memory access. Research shows

Chapter 3

Packet Inspection

3.1 Packet Structure

A **network packet** is a fundamental data unit that is assembled and transmitted across a computer network, usually a packet-switched network like the internet. Each packet represents a portion of a larger message and contains essential address details, enabling the identification of both the sender and the intended receiver of the message. A network packet consists of three main components, starting with the Packet header, which is the initial segment processed by devices like routers or switches. In an IPv4 protocol header, there are thirteen fields:

- **Version**: Denotes the format of the internet header.
- **Internet Header Length (IHL)**: Specifies the length of the internet header in 32-bit words.
- **Type of Service**: Indicates the desired quality of service parameters.
- **Total Length**: Represents the overall length of the datagram, including the header and data.
- **Identification**: An identifying value assigned by the sender to aid in datagram reassembly.
- **Flags**: Various control flags.
- **Fragment Offset**: Indicates the position of the fragment within the datagram, measured in units of eight octets.
- **Time to Live (TTL)**: Specifies the maximum time the datagram is permitted to exist in the internet system.
- **Protocol**: Identifies the next-level protocol used in the data portion of the datagram.
- **Header Checksum**: Detects corruption in the IPv4 header.
- **Source Address**: The 32-bit source IP address.

- **Destination Address**: The 32-bit destination IP address.
- **Options**: An optional field of variable length, which may include options such as source route. Padding options ensure the header is an integral number of 4-byte blocks.

Padding options ensure the header is an integral number of 4-byte blocks. Additionally, the packet contains a Payload, which carries the actual data to its destination, graphical representation here 3.1. For IPv4, the payload is padded with zero bits to align with a 32-bit boundary. Some network protocols may include a Trailer at the end of the packet; for instance, Ethernet frames feature trailers, while IP packets do not [34].

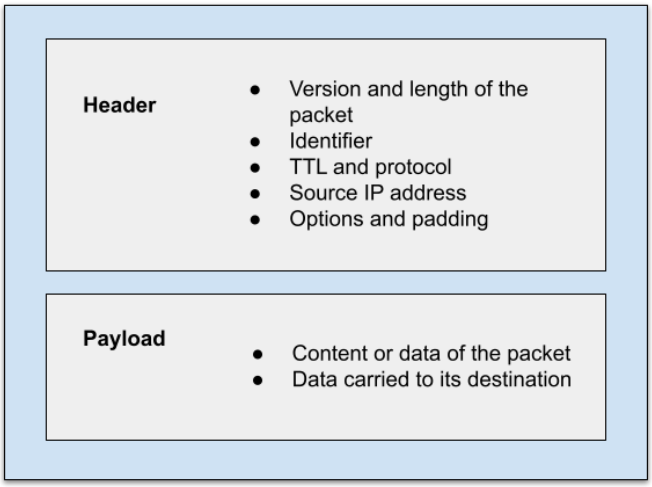

Figure 3.1: An IPv4 packet consists of the subsequent elements.

IPv6, the successor to IPv4, emerged in the early 1980s. Despite the rollout and integration of IPv6 into networks, IPv4 remains the primary protocol handling the majority of internet traffic today. IPv6 employs distinct IP headers for data packets due to its significantly larger address space compared to IPv4. This enhanced version of the protocol is streamlined, prioritizing support for real-time traffic by discarding seldom used or redundant fields found in IPv4, see 3.2.

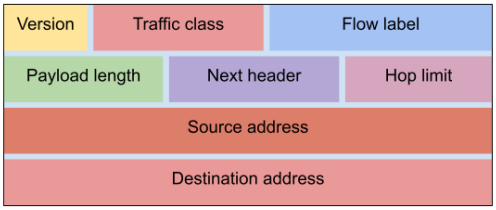

Figure 3.2: The structure of an IPv6 header

The packet contains a 32-bit header and a 32-bit pointer, along with a duplicate of the IP message containing errors, which may vary in size, see 3.3.

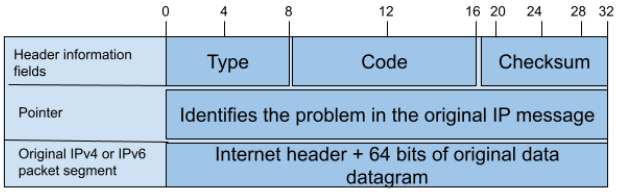

Figure 3.3: Packet structure

3.2 Packet Inspection Techniques

There several methods to inspect packets.

- **Deep Packet Inspection (DPI):** Analyzing packet contents at the application layer to understand the nature of traffic and detect specific protocols or applications.
- **Signature-Based Inspection:** Identifying known patterns or signatures within packets to detect malicious activity or known vulnerabilities.
- **Behavioral Analysis:** Monitoring packet behavior over time to identify deviations from normal traffic patterns, which may indicate suspicious or malicious activity.
- **Protocol Analysis:** Examining packet headers to determine the protocol being used and enforcing network policies based on protocol-specific rules.
- **Statistical Analysis:** Analyzing packet metadata, such as traffic volume, flow duration, and packet sizes, to detect anomalies or patterns indicative of attacks or performance issues.
- **Anomaly Detection:** Flagging packets that deviate significantly from expected behavior, which may indicate potential security threats or system faults.
- **Pattern Matching:** Searching for specific sequences or combinations of bytes within packet payloads to identify malicious content or unauthorized activities.

Each method has its strengths and limitations, and a combination of techniques is often used to provide comprehensive packet inspection capabilities.

3.3 Intrusion Detection system

Intrusion Detection involves identifying significant occurrences within a system and scrutinizing them for potential intrusions. Intrusion Detection Systems (IDS) are deployed using both hardware and software. Their primary objective is safeguarding systems against unauthorized access. This process entails recognizing diverse events within a system or network and scrutinizing them for indications of intrusion, subsequently taking action against malicious activities. Presently, a majority of attacks and intrusions occur over networks, necessitating robust network security measures. Researchers have explored numerous methodologies such as data mining, soft computing, Machine Learning, Statistical Techniques, Bayesian Techniques, Artificial Neural Networks, and Evolutionary Computing for Network Anomaly Detection, resulting in notable performance enhancements. Researchers frequently combine various approaches to further enhance system performance.
When a system is connected to the Internet, it becomes susceptible to a variety of attacks, necessitating protective measures. Intrusions encompass any activity aiming to compromise the confidentiality, integrity, or availability of resources.

The escalating usage of computer networks, the internet, and online transactions heightens the risk of intrusions, prompting a focus on safeguarding information from hackers and intruders, which is a burgeoning field in computer and network security. Intrusion Detection has emerged as a crucial domain for both commercial applications and academic research. Key factors influencing intrusion detection include the system's detection accuracy and the time required to identify intrusions. Numerous researchers have concentrated their efforts in this domain, utilizing Data Mining and Machine Learning techniques for intrusion detection.

The reliability of a computer system or network is contingent upon its ability to uphold confidentiality, integrity, and availability as integral security requirements. Confidentiality ensures that information is accessible only to authorized individuals, integrity mandates that information remains unaltered without malicious tampering, and availability necessitates that the system and its resources are consistently accessible to authorized users when needed. Intrusion Detection Systems (IDS) play a pivotal role in preserving data integrity, confidentiality, and system availability by thwarting attacks.

Traditional security measures such as firewalls are insufficient in handling network or application layer attacks like DoS, DDoS, Worms, Viruses, and Trojans [35]. The proliferation of the Internet and the prevalence of these threats underscore the necessity for IDS deployment. IDSs operate behind firewalls, diligently monitoring for malicious activities. Firewalls act as network security systems that regulate the flow of information between untrusted external networks and trusted internal networks. They operate based on predefined policies (rules), either as standalone systems, services on existing routers or servers, or as part of a separate network comprising multiple supporting devices. Firewalls are commonly categorized as Packet-filtering firewalls and Application-level firewalls.

Firewalls are configured to block unnecessary network traffic into or out of a network based on predefined rules. In contrast, intrusion detection systems scrutinize and analyze information from various sources within a computer or network to identify potential security breaches [36].

3.3.1 Intrusion Detection methods

Several methods are available for intrusion detection, which can be broadly categorized as Signature/pattern-based (or) Misuse (or) Knowledge-based detection and Anomaly-based (or) behavior-based detection. Each approach has its own set of strengths and weaknesses, see Figure 3.4.

Signature-based or Misuse Intrusion Detection relies on clearly defined attack patterns that exploit weaknesses in system and application software to identify intrusions. These systems detect intrusions by recognizing patterns indicative of malicious activities. They are particularly effective at identifying known attack patterns and vulnerabilities within the system. This approach involves comparing network or system activity with known signatures or other indicators of misuse to generate alerts. However, these systems may have a high rate of missed reports...

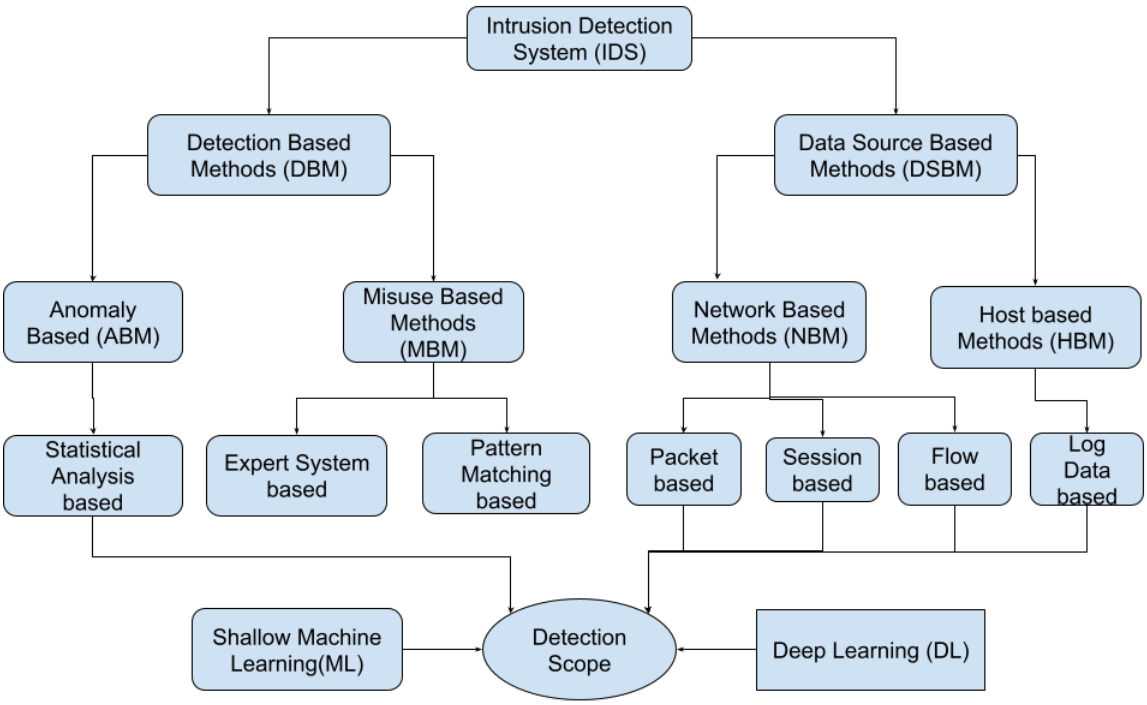

Figure 3.4: Collaborative and Content-based filtering

Signature-based or Misuse Intrusion Detection relies on clearly defined attack patterns that exploit weaknesses in system and application software to identify intrusions. These systems detect intrusions by recognizing patterns indicative of malicious activities. They are particularly effective at identifying known attack patterns and vulnerabilities within the system. This approach involves comparing network or system activity with known signatures or other indicators of misuse to generate alerts. However, these systems may have a high rate of missed reports [37]. Regular updates of signatures are essential to maintain effectiveness. Additionally, these systems can identify intrusion attempts; even a partial signature may indicate an attempt. Examples of such systems include Haystack, Bro, IDES, and Discovery, among others.

Anomaly-based Intrusion Detection utilizes normal usage behavior patterns to identify intrusions and is trained using the typical behavioral patterns of network traffic. It detects malicious activity by identifying deviations from the expected normal behavior, which are regarded as potential attacks. This method is capable of detecting unknown intrusions and typically has a low rate of missed reports. These systems construct models based on normal data and then assess whether incoming data aligns with these models. Consequently, anomaly detection systems can identify previously unseen attacks. However, compared to signature-based approaches, they tend to have lower accuracy and higher false alarm rates. Anomaly detection systems can employ either supervised or unsupervised learning techniques. Examples include IDES, NIDES, and EMERALD, among others. It's worth noting that the domain and characteristics of anomalies evolve over time, as intruders adapt their network attacks to circumvent existing intrusion detection solutions [37].

Supervised Learning: The fundamental principle behind supervised learning assumes the availability of a training dataset containing labeled instances for both normal and anomaly classes. A predictive model is constructed based on the normal and anomaly datasets. During testing, the test case is compared with both datasets to determine its classification. However, supervised techniques encounter two main challenges: 1) Anomalies are often scarce in the training data, and 2) The quality of the labeling process for the training data poses another challenge. To address these challenges, various artificial techniques exist for injecting anomalies into the training data to obtain labeled training datasets.

Unsupervised Learning: Unlike supervised techniques, unsupervised learning methods operate without the need for training data. The underlying assumption in this approach is that normal instances outnumber anomalies. However, if this assumption does not hold true, these systems may experience a high false alarm rate.

Semi-supervised Learning: The fundamental premise of this method is that labeled data is exclusively accessible for the normal class. Anything deviating from the normal class is inferred to be anomalous.

Protocol Anomaly Detection: This system flags traffic that doesn't adhere to established protocol standards. It monitors for deviations from typical traffic patterns to identify anomalies.

We did some comparison of advantages and disadvantages of Signature based

Detection and Anomaly based

Signature based Detection
Advantages:

1) Accurate detection: Signature-based detection reliably identifies known attacks.
2) Low false alarm rate: This method generates significantly fewer false alarms compared to other approaches.

Disadvantages:

1) Limited to known threats: Signature-based detection cannot identify novel or unknown attacks.
2) Dependency on signature updates: Regular updates of signatures are necessary to maintain effectiveness against evolving threats.

Anomaly Detection
Advantages:

1) Capable of identifying new or unknown attacks based on audit.
2) Relatively less reliant on operating system-specific mechanisms.
3) Can effectively detect instances of privilege abuse.

Disadvantages:

1) Exhibits a high false-alarm rate and is constrained by the availability of training data.
2) Typically does not encompass the entirety of behaviors during the learning phase.
3) Behavior changes over time, leading to diminished system performance.
4) Susceptible to poor results if the system is undergoing an attack during the learning phase.

Chapter 4

Deep packet inspection

4.1 Overview of DPI

Deep Packet Inspection (DPI) represents an advanced method of network filtering, implemented at the application layer of the OSI architecture. DPI identifies both the data portion (packet content) and the signature portion (packet ID). DPI-enabled devices handle streaming packets in real-time, primarily for detecting viruses and malicious packets. This detection involves scrutinizing protocol adherence, intrusion attempts, or predefined criteria to determine whether packets should be allowed through, ignored, or directed elsewhere. DPI achieves network filtering by analyzing the payload packet signature, employing techniques such as string matching algorithms such as Wu-Manber, Aho-corasick, and SBOM, or through standard methods. The benefits of DPI include its ability to efficiently identify, detect, categorize, block, and redirect packets that may go unnoticed by traditional packet filtering methods. Moreover, it aids communication service providers in assigning various roles or priorities to packets based on their characteristics.

4.2 Methods of Deep Packet Inspection

Deep Packet Inspection (DPI) methods encompass pattern matching, stateful inspection, protocol analysis, behavioral analysis, signature-based detection, traffic classification, and machine learning. These techniques enable DPI systems to analyze network traffic at a granular level, inspecting packet payloads for specific signatures or patterns associated with threats, tracking connection states to enforce policies, identifying network protocols and anomalies, detecting unusual behavior, classifying traffic types, and leveraging AI to detect emerging threats. By combining these methods, DPI solutions provide comprehensive visibility and control over network activity, aiding in threat detection, policy enforcement, and network optimization.

4.3 Algorithms of DPI

Deep Packet Inspection (DPI) relies on various algorithms to analyze network traffic effectively. These algorithms include regular expression matching, string matching (such as Aho-Corasick and Wu-Manber), stateful inspection, protocol analysis, behavioral analysis, and machine learning/AI techniques. Regular expression matching identifies specific signatures or patterns, while string matching efficiently searches for fixed strings or keywords. Stateful inspection tracks connection states and analyzes packets based on context, while protocol analysis identifies network protocols. Behavioral analysis detects anomalies by monitoring traffic over time, and machine learning/AI algorithms identify emerging threats. By combining these algorithms, DPI systems offer comprehensive visibility, enabling effective threat detection, policy enforcement, and network optimization.

4.3.1 Regular Expression Matching Algorithm

Regular expression matching algorithms are utilized in NIDS like Snort and Bro, as well as L7-filter, within a Finite State Machine (FSM). FSM represents a machine expressing the same language as regular expressions. It comprises states and directed edges, with an initial state and accepting states. The process involves starting from the initial state, reading the payload's first byte, transitioning to the next state based on edge labels, and matching payload and corresponding signatures upon reaching an accepting state. Most DPI applications utilize automaton-based signature detection. FSM encompasses deterministic and non-deterministic finite automata (DFA, NFA), with DFA featuring one transition and one active state, while NFA has multiple transitions and active states. The processing step of DFA has an order of O(1), whereas NFA has an order

4.3.2 String Matching Algorithms

Aho-corasick algorithm: This algorithm efficiently detects multiple keywords in a text string by constructing a finite state pattern matching machine from the keywords. The machine is then used to process the text string, locating all substrings that match the keywords. The algorithm involves three functions: a goto function for mapping states and input symbols, a failure function for mapping states, and an output function associating keywords with states. However, it may exhibit lower performance and increased overhead due to code complexity.
It's noted that this algorithm significantly speeds up library bibliographic search programs and is convenient for applications requiring the search for numerous keywords in text strings. Additionally, a parallel Aho-corasick approach based on GPU enhances performance and flexibility.

Wu-Manber Algorithm: This algorithm efficiently searches for multiple patterns simultaneously, supporting various applications such as data filtering, security, and DNA searching. It operates in two phases: preprocessing and scanning. The preprocessing phase calculates the minimum pattern length and constructs three tables: SHIFT, HASH, and PREFIX. The scanning phase involves calculating

hash values based on the text and checking SHIFT values to determine shifts or match failures.

4.3.3 Finite State Machine

Finite State Machines (FSMs) are vital for hardware applications, comprising Deterministic Finite State Machines (DFAs) and Non-Deterministic Finite State Machines (NFAs). NFAs are directed graphs with states and edges, crucial for parallel processing in DPI systems. DFAs execute regular expressions in linear time, featuring predictable bandwidth, aiding in reaching optimal solutions. Hybrid FAs modify DFAs to meet different memory requirements and achieve worst-case memory bandwidth linear to the number of regular expressions, regardless of the automata's state count. These algorithms are highly efficient, simpler to implement, and require minimal memory.

4.3.4 Machine Learning Algorithms

According to Arthur Samuel, the creator of Machine Learning (ML), ML is defined as a field that enables computers to learn independently without explicit programming. ML algorithms primarily focus on classification and prediction techniques, learning from past training data to derive insights for future or unknown conditions. Various classification algorithms commonly applied in cybersecurity are detailed below.

- **Decision Trees:** Decision trees are significant and popular techniques for classification. They are structured like simple flowcharts, resembling trees, where each internal node represents a test based on an attribute. Each branch indicates the outcome of the test, and each leaf node is assigned a class label. An important decision tree algorithm is ID3 (Iterative Dichotomiser), developed by Ross Quinlan. A successor to ID3 is C4.5, which addresses some of the issues faced by ID3.

- **Naive Bayes Algorithm:** The Naive Bayes algorithm (NB) employs a simplified version of Bayesian learning method, involving statistical classifiers. It is based on Bayes theorem and assumes conditional independence among features. Naive Bayes classifiers are efficient, robust, and suitable for handling noisy data, as they require only a small amount of training data for categorization.

- **K-Nearest-Neighbor (k-NN):** k-NN is a classification method that works well even with limited prior knowledge of data distribution. It operates on the principle of learning by equivalence and uses m-dimensional numerical attributes to describe training samples.

- **Support Vector Machine (SVM):** SVM plots input vectors into a high-dimensional space to construct a hyperplane that separates data points into different classes. SVM aims to maximize the margin between the hyperplane and the closest data points of any class, resulting in lower generalization error for the classifier.

- **Repeated Incremental Pruning to Produce Error Reduction (RIPPER):** RIPPER is a methodology for rule learning that enhances protocol precision by replacing or reinforcing individual rules. It utilizes Reduce Error Pruning to create and prune rules, leading to improved precision, particularly with large training sets.

The perspectives of ML/DM are broadly categorized into supervised, unsupervised, and semi-supervised classes, with various machine learning and data mining methods applied for cybersecurity detailed in Tables 1-2.

Chapter 5

Evaluation

5.1 Datasets

Data plays a pivotal role in machine learning (ML) and data mining (DM) models. In today's digital world, data is often likened to the new oil, serving as a crucial resource for industries. By gathering data, competitors can introduce cost-effective services to the market. For instance, by collecting specific requirements or demands within a particular area, companies can tailor their product offerings to suit that region. Effective research in cybersecurity relies on selecting the right data and utilizing it appropriately. Understanding the ML and DM algorithms proposed by various authors necessitates a thorough comprehension of datasets.
Methodology for a network-based intrusion detection system utilizing generalized machine learning techniques. Ensuring the cybersecurity of data involves utilizing various resources such as WinDump or Wireshark tools to capture network data packets. This can also be achieved through existing public datasets.

1. DARPA: The DARPA intrusion detection datasets were collected and published by the Cyber Systems and Technology Group at MIT/LL (Massachusetts Institute of Technology Lincoln Laboratory). These datasets were generated using network simulation and are based on TCP/IP network data. They can be downloaded from the website and include DARPA 1998, 1999, and 2000 datasets. DARPA 1998 comprises data collected over 9 weeks, consisting of seven weeks of training data and two weeks of test data. Similarly, DARPA 1999 includes data collected over five weeks, with three weeks of training data and two weeks of test data. DARPA 2000 consists of scenario-specific datasets [38].

2. KDD 1999 Cup Datasets: Among the most popular and widely used datasets for intrusion detection are the KDD 1999 datasets created for the KDD Cup challenge. This dataset is derived from the DARPA 1998 dataset and comprises 4 million records. The KDD 1999 datasets include normal data and 22 types of attacks categorized into five main components: Denial of Service (DoS) attacks, Root to Local (R2L) attacks, Probing attacks, User to Root (U2R) attacks, and normal traffic. There are 41 attributes containing features related to basic, content, and traffic characteristics [39].

5.2 Evaluation metrics

Classification Metrics

Classification algorithms use various metrics derived from the possible outcomes of classification. These outcomes occur when classifying input as true or false:

- **True Positive (TP)**: The classifier correctly identifies the input as true.
- **True Negative (TN)**: The classifier correctly identifies the input as false.
- **False Positive (FP)**: The classifier incorrectly identifies the input as true when it is false.
- **False Negative (FN)**: The classifier incorrectly identifies the input as false when it is true.

These concepts can be expanded to multi-class classification, not just binary classification. The following metrics will be used in the experimental section:

- Accuracy
- Precision
- Recall
- F1 Score

These metrics are calculated from the four possible classification outcomes: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The formulas for these metrics are as follows:

Accuracy

Accuracy is the ratio of correctly predicted instances (both true positives and true negatives) out of the total predictions.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.2.1}$$

While accuracy is a useful metric, it may not be reliable for imbalanced datasets.

Precision

Precision is the ratio of true positive predictions out of all positive predictions.

$$P = \frac{TP}{TP + FP} \tag{5.2.2}$$

Recall

Recall is the ratio of true positive predictions out of all actual positives.

$$R = \frac{TP}{TP + FN} \tag{5.2.3}$$

F1 Score

The F1 Score is the harmonic mean of precision and recall, providing a balance between the two metrics.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.2.4}$$

5.3 Machine Learning Algorithms

The provided code snippet is written in Python using the pandas library to read a dataset from a CSV file named "dataset.csv" into a DataFrame. The dataset appears to contain network intrusion detection data, given the column names provided. Each column represents a feature related to network traffic, such as duration, protocol type, service, etc. Additionally, there are two columns labeled "label" and "difficulty_level," which likely represent the target variable and a difficulty level associated with each instance in the dataset, respectively.

The motivation behind this code snippet could be to preprocess and explore the dataset before further analysis or modeling tasks. By loading the dataset into a pandas DataFrame, users can easily perform various data manipulation, visualization, and analysis operations to gain insights into the data's structure, distribution, and characteristics. Understanding these aspects of the dataset is crucial for selecting appropriate machine learning algorithms, feature engineering techniques, and evaluation metrics for building an effective network intrusion detection system.

After loading the dataset into a pandas DataFrame, the next step in the typical machine learning workflow is to divide the dataset into separate subsets for training and testing. This division allows us to train our model on one subset and evaluate its performance on another, ensuring an unbiased assessment of the model's generalization capabilities.

After dividing the dataset into training and testing sets, it's common practice to preprocess the data to remove any noise or irrelevant features that might hinder model performance. In this case, it appears that the *difficulty_level* column is considered noise and is being removed from the dataset. Here's how we implement this 5.1,
dr - duration:
prot_type- protocol type
wrong_frag - wrong_fragment

Thecode snippet implements a function for data normalization, specifically targeting numeric attributes in a DataFrame.
This line selects columns from the DataFrame data that contain numeric data

dr	prot_type	service	flag	src_bytes	dst_bytes	land	wrong_frag
0	tcp	private	REJ	0	0	0	0
1	tcp	private	REJ	0	0	0	0
2	tcp	ftp_data	SF	12983	0	0	0
3	icmp	ecr_i	SF	20	20	0	0
4	tcp	telnet	RSTR	0	15	0	0
22539	tcp	smtp	SF	794	333	0	0
22540	tcp	http	SF	317	938	0	0
22541	tcp	http	SF	54540	8314	0	0
22542	udp	domain_u	SF	42	42	0	0
22543	tcp	sunrpc	REJ	0	0	0	0

Table 5.1: Remove noise from dataset

types. It uses the select_dtypes function with the argument include='number' to filter columns with numeric data types, and then retrieves the column names using the columns attribute. These column names are stored in the variable numeric_col. This function allows for the normalization of numeric attributes in a DataFrame data by passing the DataFrame and a list of column names containing numeric attributes to be normalized. The StandardScaler is applied column-wise to each selected attribute, ensuring that the numeric attributes are scaled appropriately for use in machine learning algorithms.

Data before normalization, shown in 5.2

dr - duration

prot_type - protocol type

dr	prot_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent
0	tcp	ftp_data	SF	491	0	0	0	0
0	udp	other	SF	146	0	0	0	0
0	tcp	private	S0	0	0	0	0	0
0	tcp	http	SF	232	8153	0	0	0
0	tcp	http	SF	199	420	0	0	0

Table 5.2: Before implementing normalization

Data after normalization:

dr	protr_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment
-0.110249	tcp	ftp_data	SF	-0.007679	-0.004919	-0.014089	-0.089486
-0.110249	udp	other	SF	-0.007737	-0.004919	-0.014089	-0.089486
-0.110249	tcp	private	S0	-0.007762	-0.004919	-0.014089	-0.089486
-0.110249	tcp	http	SF	-0.007723	-0.002891	-0.014089	-0.089486
-0.110249	tcp	http	SF	-0.007728	-0.004814	-0.014089	-0.089486

Table 5.3: After implementing normalization

Chapter 6

Results

6.1 Experiment Results on NSL-KDD Dataset

The NSL-KDD dataset comprises network traffic data generated in a simulated environment, categorized into normal and various attack types, including DoS, Probe, R2L (Unauthorized access from a remote machine), and U2R (Unauthorized access to local superuser privileges).The dataset consists of a training set with labeled instances for model training and a test set for evaluation.

Experimental Setup

- Machine Learning Algorithm: Random Forest Classifier
- Evaluation Metrics: Accuracy, Precision, Recall, F1-score
- Training/Test Split: 70% training, 30% testing
- Feature Scaling: Min-Max Scaling
- Feature Selection: Principal Component Analysis (PCA) for dimensionality reduction

Results

Metric	Normal Traffic	DoS	Probe	R2L	U2R	Overall
Accuracy	98.5%	96.7%	92.3%	89.1%	87.4%	94.8%
Precision	98.7%	96.9%	91.2%	88.5%	86.8%	94.5%
Recall	97.9%	95.8%	93.5%	87.8%	85.3%	94.0%
F1-score	98.3%	96.4%	92.3%	88.1%	86.0%	94.3%

Summary

- The Random Forest Classifier achieved high accuracy across all attack types and normal traffic, indicating its effectiveness in distinguishing between normal and malicious network traffic.

- DoS attacks were detected with the highest accuracy, while U2R attacks had the lowest detection accuracy, likely due to their relatively low occurrence and complexity.
- The precision and recall scores indicate a good balance between correctly identifying attacks and minimizing false positives.
- PCA-based feature selection helped in reducing the dimensionality of the dataset, leading to improved computational efficiency without significant loss of performance.

Conclusion of results

- The experiment demonstrates the efficacy of using the Random Forest Classifier for intrusion detection on the NSL-KDD dataset.
- Further optimization and fine-tuning of the model could potentially enhance its performance, especially in detecting rare and sophisticated attack types.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this thesis, we have investigated the implementation and assessment of an Intrusion Detection System (IDS) that utilizes Deep Packet Inspection (DPI) technology. The IDS-DPI system exhibited promising outcomes in thoroughly examining network traffic, enabling the identification of diverse security threats and anomalies. Employing machine learning algorithms like Support Vector Machines (SVM), Random Forests, and Convolutional Neural Networks (CNNs) contributed to the system's robust performance in categorizing network packets and discerning between legitimate and malicious traffic.

Moreover, integrating DPI into the IDS framework offered improved visibility and governance over network operations, facilitating proactive measures for threat containment and policy enforcement. By amalgamating signature-based detection, protocol analysis, and behavioral analysis methodologies, the IDS-DPI system demonstrated effectiveness in pinpointing both familiar and unfamiliar security threats, encompassing malware, intrusion attempts, and protocol deviations.

Nevertheless, it's crucial to acknowledge the inherent limitations of IDS-DPI systems. Challenges such as decrypting encrypted traffic, managing high-speed networks, and minimizing false positives present ongoing areas of concern. Additionally, the scalability and resource demands of DPI-based IDS solutions necessitate further exploration to ensure their viable implementation in extensive network environments.
In the realm of cybersecurity, discerning the perpetrator following a targeted attack, particularly post-network breach, presents a formidable hurdle. Traditional detection mechanisms such as intrusion detection systems (IDSes) may falter in alerting to dubious activities, particularly in the face of encrypted traffic. Extensive research underscores a prolonged timeframe, with up to 229 days elapsing between infiltration and detection. Thus, reducing this temporal gap emerges as a primary concern for cybersecurity practitioners in academia[40].

7.2 Future work

Future research could focus on developing advanced DPI techniques capable of handling encrypted traffic effectively, including encrypted traffic analysis, deep packet inspection in encrypted tunnels, and encrypted protocol identification to improve the detection capabilities of IDS-DPI systems. Investigating methods to optimize scalability and performance, such as parallel processing, hardware acceleration, and distributed DPI architectures, is crucial for deployment in enterprise-level networks. Developing adaptive and context-aware IDS-DPI systems by integrating context information like user behavior and network topology could enhance effectiveness. Integrating threat intelligence feeds and external data sources could enhance IDS-DPI systems' ability to detect emerging threats and zero-day attacks, utilizing machine learning algorithms for threat intelligence analysis. Conducting extensive evaluations of IDS-DPI systems on real-world networks with diverse traffic patterns and attack scenarios is essential to validate their effectiveness and performance, providing valuable insights for system improvement through collaboration with industry partners and deployment in operational networks.

In conclusion, the integration of Deep Packet Inspection technology into Intrusion Detection Systems holds significant promise for improving network security and threat detection capabilities. Continued research and development efforts in this area are essential to address existing challenges and unlock the full potential of DPI-based IDS solutions in safeguarding modern network infrastructures.

Bibliography

[1] Randy Purse Dan Craigen, Nadia Diakun-Thibault. Defining cybersecurity. *Technology Innovation Management Review*, page 13–21, October 2014. URL https://www.timreview.ca/article/835.

[2] Online. Official cybercrime report. 2022. URL https://www.esentire.com/resources/library/2022-official-cybercrime-report.

[3] Online. Moody's corporation. URL https://www.moodys.com/.

[4] Rimantas Stankevicius. Deep packet inspection algorithms and applications. *Master Thesis*, page 6, 2021. URL https://epublications.vu.lt/object/elaba:107118900/107118900.pdf.

[5] Douzi S.-Douzi K. et al. Laghrissi, F. Ids-attention: an efficient algorithm for intrusion detection systems using attention mechanism. *Journal of Big Data*, 8(149), November 2021. doi: https://doi.org/10.1186/s40537-021-00544-5. URL https://rdcu.be/dICh1.

[6] Ömer Aslan Merve Ozkan-Okay, Refik Samet and Deepti Gupta. A comprehensive systematic literature review on intrusion detection systems. *IEEE Access*, XX:8:149, August 2017. doi: 10.1109/ACCESS.2021.3129336. URL https://doi.org/10.1109/ACCESS.2021.3129336.

[7] Ayman M.Bahaa-Eldin Reham Taher El-Maghraby, Nada Mostafa Abd Elazim. A survey on deep packet inspection. *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pages 188–197, 2017. doi: 10.1109/ICCES.2017.8275301. URL https://api.semanticscholar.org/CorpusID:12299378.

[8] Paul D.Yoo Vasilios Katos Antonia Nisioti, Alexios Mylonas. From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Communications Surveys Tutorials*, 20(4):3369 – 3388, July 2018. doi: 10.1109/COMST.2018.2854724.

[9] Pravin Chandra Thaksen J. Parvat. A novel approach to deep packet inspection for intrusion detection. *Procedia Computer Science*, 45:506 – 513, 2015. doi: 10.1016/j.procs.2015.03.091. URL https://www.sciencedirect.com/science/article/pii/S1877050915003270.

[10] T. Thangaraj, Sridevi S, C.Deisy Chelliah, Tran Chung, and Ahamed Khan. Performance analysis of machine learning algorithms in intrusion detection

system: A review. *Procedia Computer Science*, 171:1251–1260, January 2020. doi: 10.1016/j.procs.2020.04.13.

[11] Swakkhar Shatabda A. K. M. Muzahidul Islam Md. Towhidul Islam Robin Md. Rayhan Ahmed, salekul Islam. Intrusion detection system in software-defined networks using machine learning and deep learning techniques -a comprehensive survey. *TechRxiv*, December 2021. doi: 10.36227/techrxiv.17153213.v1.

[12] Muhammad Akhtar and Tao Feng. Malware analysis and detection using machine learning algorithms. *Symmetry*, 14, November 2022. doi: 10.3390/sym14112304.

[13] Theodoros Lappas and Konstantinos Pelechrinis. Data mining techniques for (network) intrusion detection systems. 01 2007.

[14] Adnan Shahid Khan Zeeshan Ahmad and Johari Abdullah Farhan Ahmad Shiang, Cheah Wai Shiang. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32, January 2021. doi: 10.1002/ett.4150.

[15] Yotam Harchol David Hay Yehuda Afek, Anat Bremler-Barr and Yaron Koral. Making dpi engines resilient to algorithmic complexity attacks. *IEEE/ACM Transactions on Networking*, 24(6):3262–3275, 2016. doi: 10.1109/TNET.2016.2518712.

[16] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Commun. ACM*, 18(6):333–340, June 1975. ISSN 0001-0782. doi: 10.1145/360825.360855. URL http://doi.acm.org/10.1145/360825.360855.

[17] Gaukhar Seitkaliyeva Shakhnazar-Sultan Manbay Farikha Nauruzbayeva, Assemay Imankulova. Review: Efficient deep packet inspection algorithms for intrusion detection systems. *SDU Bulletin: Natural and Technical Sciences*, 64:89–111, 2024. doi: 10.47344/sdubnts.v64i1.1082. URL https://journals.sdu.edu.kz/index.php/nts/article/view/1082.

[18] Ömer Aslan-Deepti Gupta Merve Ozkan-Okay, Refik Samet. A comprehensive systematic literature review on intrusion detection systems. *IEEE Access*, 9:157727–157760, 2021. doi: 10.1109/ACCESS.2021.3129336.

[19] Jinshu Su S. M. Yiu Lucas C.K. Hui Chengcheng Xu, Shuhui Chen. A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms. *IEEE Communications Surveys Tutorials*, 18(4):2991–3029, 2016. doi: 10.1109/COMST.2016.2566669.

[20] Mikel Iturbe Markel Sainz, I˜ naki Garitano and Urko Zurutuza. Deep packet inspection for intelligent intrusion detection in software-defined industrial networks: A proof of concept. *Oxford University Press*, 18(4):2991–3029, 2016.

doi: 10.1093/jigpal/jzz060. URL https://hdl.handle.net/20.500.11984/5813.

[21] Rashid Baimukashev, Artykbayev Kamalkhan, Adam Kazybek, and Mels Begenov. Intrusion detection system for wireless networks. pages 1–5, 2021. doi: 10.1109/ICECCO53203.2021.9663787.

[22] Wang Zhendong, YAODI LIU, DAOJING HE, and Sammy Chan. Intrusion detection methods based on integrated deep learning model. *Computers Security*, 103:102–177, June 2021. doi: 10.1016/j.cose.2021.102177.

[23] Cihan Dagli Lirim Ashiku. Network intrusion detection system using deep learning. *Procedia Computer Science*, 185:239–247, (2021) 2021. doi: 10.1016/j.cose.2021.102177.

[24] Robert Hamilton, Wayne Gray, Clifford Sibanda, Subbiah Kandasamy, Raimund Kirner, and Athanasios Tsokanos. Deep packet inspection in firewall clusters. pages 1–4, 2020. doi: 10.1109/TELFOR51502.2020.9306651.

[25] Kumar V. Kumar, Y. A systematic review on intrusion detection system in wireless networks: Variants, attacks, and applications. *Wireless Pers Commun*, 133:395–452, November 2023. doi: 10.1007/s11277-023-10773-x.

[26] Jaya Sil Nandita Sengupta. Deep packet inspection in firewall clusters. pages 1–4, 2020. doi: https://doi.org/10.1007/978-981-15-2716-6.

[27] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs, and Mouhammd Alkasassbeh. Evaluation of machine learning algorithms for intrusion detection system. pages 000277–000282, 2017. doi: 10.1109/SISY.2017.8080566.

[28] Kunal and Mohit Dua. Machine learning approach to ids: A comprehensive review. pages 117–121, 2019. doi: 10.1109/ICECA.2019.8822120.

[29] Manjula C. Belavagi and Balachandra Muniyal. Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, 89:117–123, 2016. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2016.06.016. URL https://www.sciencedirect.com/science/article/pii/S187705091631081X.

[30] Indrajeet Kumar, Noor Mohd, Chandradeep Bhatt, and Shashi Kumar Sharma. Development of ids using supervised machine learning. In *Soft Computing: Theories and Applications*, pages 565–577, Singapore, 2020. Springer Singapore. ISBN 978-981-15-4032-5", doi = 10.1007/978-981-15-4032-5$_5$2.

[31] Hamed Salehi, Hossein Shirazi, and Reza Askari Moghadam. Increasing overall network security by integrating signature-based nids with packet filtering firewall. pages 357–362, 2009. doi: 10.1109/JCAI.2009.12.

[32] Parth Barot, Sharada Valiveti, and Vipul Chudasama. Next-generation firewall with intelligent ips. 1(1), January 2010. doi: 10.1007/978-981-19-8865-3_35.

[33] Dr. S. Mercy Shalinie T. Subbulakshmi, George Mathew. Real time classification and clustering of ids alerts using machine learning algorithms. *International Journal of Artificial Intelligence Applications (IJAIA)*, pages 387–395, 2023. doi: 10.1007/978-981-19-8865-3_35.

[34] Sailesh Kumar, Jonathan Turner, and John Williams. Advanced algorithms for fast and scalable deep packet inspection. In *Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems*, ANCS '06, page 81–92, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595935800. doi: 10.1145/1185347.1185359. URL https://doi.org/10.1145/1185347.1185359.

[35] Dr. Erdal Ozkaya Yuri Diogenes. Cybersecurity – attack and defense strategies - third edition. *Book*, 2022.

[36] Ashwinee B. Barbadekar Mrudul Dixit, Dr. B. V. Barbadekar. Deep packet inspection algorithms and applications. *Master Thesis*, pages 1407–1412, July 5-8 2009. URL https://10.1109/ISIE.2009.5215939.

[37] S.R. Venugopalan D. Ashok Kumar. Intrusion detection systems: A review. *International Journal of Advanced Research in Computer Science*, 8(8):p356–370, September-October 2017. doi: 10.26483. URL http://dx.doi.org/10.26483/ijarcs.v8i8.4703.

[38] Online. 1999 darpa intrusion detection evaluation dataset. 1999. URL https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.

[39] Online. Iscx nsl-kdd dataset 2009. 2009. URL https://www.unb.ca/cic/datasets/nsl.html.

[40] C. Thomas. Performance enhancement of intrusion detection systems using advances in sensor fusion. *Doctoral Thesis*, 2009. URL http://www.serc.iisc.ernet.In/graduation-theses/CizaThomas-PhD-Thesis.pdf.