AI-Enhanced Vulnerability Detection: A Machine Learning Approach to Automated Penetration Testing

by

Mahmud Mostofa Al Maruf
22101132
Mahmuda Tabassum
22101151
Radiah Reaz Disha
22101545
Salauddin Ahmed Emon
24141192

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October 2025.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

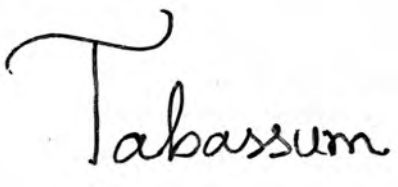
Student's Full Name & Signature:

Mahmud Mostofa Al Maruf

22101132

Salauddin Ahmed Emon

24141192



Mahmuda Tabassum

22101151

Radiah Reaz Disha

22101545

Approval

The thesis/project titled "AI-Enhanced Vulnerability Detection: A Machine Learning Approach to Automated Penetration Testing" submitted by

1. Mahmud Mostofa Al Maruf (22101132)
2. Mahmuda Tabassum (22101151)
3. Radiah Reaz Disha (22101545)
4. Salauddin Ahmed Emon (24141192)

of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in Fall 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain

Associate Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Cybersecurity is an issue that is being compounded by the fast development of digital infrastructure as cyberattacks grow more sophisticated and frequent. Vulnerability detection and penetration testing are some of the most important challenges in the area of cybersecurity as they allow organizations to highlight the security vulnerabilities prior to the exploitation. The existing traditional penetration testing is, however, largely manual, expensive, inefficient, and time-consuming, especially in large scale dynamic networks. This paper presents an improved vulnerability detection system using AI that can be used to automate penetration testing to enhance the ability of the system to detect threats more effectively. It incorporates automated attack path exploration, AI based threat intelligence and adversarial defense to increase the precision, versatility and dependability of penetration tests. To make security assessment transient and interpretable, explainable-AI methods are also included. Findings show that AI-based penetration testing also saves considerable time in human work, and it detects high-risk threats more efficiently through the analysis of large datasets, including complicated attack patterns. This piece is emphasizing a change to the conventional way of approaching cybersecurity as it provides a more effective and scalable alternative to the conventional way of doing it.

Keywords: Artificial Intelligence, machine learning, penetration testing, automated vulnerability detection, cybersecurity, threat intelligence, AI security, attack path optimization, adversarial defense, explainable AI, security automation, vulnerability assessment, and cyber threat detection.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Cybersecurity is an aspect that has gained critical concern in the contemporary digital age driven by the fact that the systems and data-driven operations being the foundation of global economies, governments and societies are interconnected. The discovery of new technologies, including cloud computing, the Internet of Things (IoT), artificial intelligence, and remote working facilities, have increased the attack surface exponentially and organizations are more vulnerable to cyber threats than ever. Vulnerabilities- weaknesses or vulnerabilities in software, hardware, networks or processes are points of entry by malicious actors to execute attacks, including breaches of data, ransomware but also espionage and disruption of a denial of service. Nearly long, conventional penetration testing has been an element of cybersecurity defenses. This is an active approach whereby ethical hackers are employed to mimic real life attacks in order to identify and rectify vulnerabilities. Nevertheless, such an approach is frequently constrained due to its highly manual nature, which demands a great deal of expertise and time as well as resources. Consequently, the rate of testing can be reduced, and the reaction to the threats that arise can be slow.

The introduction of Artificial Intelligence (AI), and the more specific concept of machine learning-based AI methods to the field of cybersecurity activities is a radical move to quicker, more precise, and more responsive security measures. The AI systems can process large amounts of data, detect even the slightest patterns, and adjust to the emerging threats more effectively than they can be in purely manual methods. As an example, ML algorithms can be used to scan the codebases to detect possible security vulnerabilities or match anti-network activity with familiar exploit signature patterns. The rising number and complexity of cyber incidents explain the rising use of such methods. It is reported that in 2024, over 30,000 additional security vulnerabilities were detected, which is 17% higher than the number of the previous year [20]. Moreover, worldwide cyber attacks also increased by 30 percent in the second quarter of 2024 with an average of 1,636 attacks per single organization per week [22]. The economic impact is also unbelievable: the average cost of a data breach in 2024 is $4.88 million, which is a 10 percent increase to the year before and, yet, it amounts to a heavy burden on the economy since it is a breach notification cost, not data recovery [19]. There were over 6.8 billion records that were

compromised in 2,741 publicly reported incidences of vulnerability in the United States alone in the course that happened between November 2023, and April 2024 [15]. Healthcare is one of the easiest victims as in 2024 alone, the shielded health information of 276 million people was disclosed or stolen, which is 758,288 records stolen daily on average [18].

The paper has started with a conceptual and theoretical analysis of how the techniques of machine learning can be used to automate the processes of vulnerability scanning and vulnerability analysis. Some of the key terms were Common Vulnerabilities and Exposures (CVEs), a standard list of publicly disclosed vulnerabilities; Common Weakness Enumerations (CWEs), a list of underlying vulnerabilities in software and hardware and frameworks like Common Attack Pattern Enumerations and Classifications (CAPECs) to describe attacker behavior. They were also discussed through predictive mappings to the MITRE ATT&CK framework that describes adversary tactics, techniques and procedures (TTPs), and D3FEND against defensive strategies. On this basis, the study was further advanced to an implementation phase where a complete pipeline was established, using network scanning to extract CVEs, XML parsing to extract CVEs, CWEs were mapped to datasets and API used, and finally, the multi-label machine learning models were used to predict. The practical case studies provided to support the validity of this methodology were carried out on various publicly available real-world websites, where machine learning-based AI methods showed the ability to improve penetration testing by making it more a continuous and smart process rather than a periodic manual one.

The biography also emphasizes the human factor in cybersecurity attacks because about 60 percent of attacks are insider-related, or human error, and 83 percent of organizations say that they have at least one incident associated with insiders in 2024 [21]. In addition, 75% of the attacks are now considered malware-free, which highlights the necessity of sophisticated and AI-supported software that would be able to trace even unobtrusive and non-conventional vectors of threats [23]. The need to scale and automate cybersecurity has become an urgent issue following the fact that cybercrime is projected to be responsible for annual damages amounting to $10.5 trillion in the years 2025, an amount that is equal to the third-largest economy in the world [24].

1.2 Rationality of the Research or Motivation

The motivation to conduct this research has its ground in the sense of urgency to respond to the growing complexity and volume of cyber threats and overcome the inefficiencies of conventional ways of cybersecurity. The estimated annual cost of cybercrime of $10.5 trillion by 2025 does not only reflect a colossal cost on the economy, but also it is a severe threat to national security, individual privacy, and even social stability [24]. Although it is considered valuable, traditional penetration testing is reactive and is not performed frequently, with a great number of organizations undertaking their assessment once a year or once in six months because it is rather expensive and consumes a significant amount of resources. This low frequency poses great defensive gaps with an average breach detection time at about 258 days and this gives the attackers a lot of room in inflicting damages.

The benefits of artificial intelligence, especially, machine learning-based AI solutions, to the penetration testing process are quite convincing and inspire the current study. Automated testing is fast in detecting the vulnerabilities as it can perform deeper analysis of the threat on a large number of data more effectively as compared to manual methods. These techniques can be implemented on large networks and complex systems, which is why they are applied to organizations of different sizes. In predictive analytics, machine learning will be capable of finding possible weaknesses before they are exploited, with automation eliminating dependence on manual procedures and minimising human mistakes. In comparison, AI-assisted penetration testing is not only faster than traditional ones, but it is also more flexible to the changes in threats, and it can also model more complex attacks based on data-driven reasoning. The capabilities enable the end-to-end assessment process to be streamlined, operational costs to be reduced, the frequency at which the tests are executed to be increased and that the coverage of security is more uniform.

Moreover, AI-based solutions can be used to democratize cybersecurity in small and medium enterprises (SMEs), which tend to have no dedicated security team and resources available to them. These systems will enable the firms to ensure effective security postures are maintained at near real-time without imposing prohibitive costs. This study is even more justified by the fact that AI-based cyberattacks have already appeared, where the attackers use machine learning algorithms to avoid being detected; to eliminate such threat, the state must also have highly developed defense systems. In that regard, the proposed research will make a contribution by adopting a system which not only detects vulnerabilities, but also forecasts the probable attack routes and suggests countermeasures which follows the trend of proactive and intelligence-oriented approaches to cybersecurity.

The timeliness of the present research is demonstrated by the increasing usage of AI-supported tools in security operations centers (SOCs) as well as the rising possibility of performing continuous penetration testing. Since the world of threat escalation is changing at an alarming rate, the need to have assessments tools with the capability of dynamically changing and giving actionable insights in a scalable and sustainable way is evident.

1.3 Problem Statement

Although automated vulnerability scanners have become very popular, the output of vulnerability assessment is often provided as long lists of Common Vulnerabilities and Exposures (CVE) identifiers. Whereas such reports show the existence of known vulnerabilities, the information is not well contextualized on how the vulnerabilities can be exploited or how these vulnerabilities can be prioritized to be mitigated. Consequently, this has led to security teams frequently needing to be able to bridge the vulnerability identification and actionable decisions on security by performing manual triage, ad-hoc analysis, and through personal experience.

This is a time-consuming, inconsistent, and scaling-challenged process of manual interpretation and is especially challenging in the face of the increasing set of disclosed vulnerabilities, on one hand, and the rapidly changing techniques of the attacker,

on the other hand. This is not the problem of not having any vulnerability data, but rather the problem of the absence of organized systems to transform raw scan data into attacker-focused and defense-focused intelligence.

In order to fill this gap, an automated and reproducible method is necessary that can convert regular evidence of vulnerability scans into implementable advice. In particular, with a list of known CVEs, a system must be capable of deducing the basic categories of weaknesses and apply this data to predict probable actions of the attacker along with the appropriate countermeasures. The main problem is associated with providing this mapping in a manner that is reliable, effective and interpretable to enable the delivery of outputs that are easily auditable and can be incorporated into the current security processes. This would help organizations plan their testing and mitigation efforts on the basis of an informed decision instead of a manual interpretation.

1.4 Objectives

The research questions in this case study aim at solving the issue of converting the raw vulnerability scan results into actionable security intelligence based on an automated and intelligible framework. The objectives of this particular research are:

- To design and develop a pipeline to extract Common Vulnerabilities and Exposures (CVEs) in XML format, through network scan outputs, to be able to accurately and reliably parsify and find unique vulnerabilities.
- To deploy the mechanism of retrieving matching Common Weakness Enumerations (CWEs) by using an updated master dataset, and fallback on the National Vulnerability Database (NVD) API to retrieve missing entries, within the framework of rate-limiting limitations.
- To implement and train multi-label machine learning classification models with the help of Random Forest classifiers to predict Common Attack Pattern Enumeration and Classification (CAPEC) patterns, MITRE ATT&CK techniques, and D3FEND defensive measures on the basis of CWE-derived features.
- In order to test the suggested system by providing actual case studies based on the real-life cases that were carried out using publicly available websites, one would have to examine the vulnerabilities identified and predictions produced in order to evaluate the practical relevance and effectiveness.
- To determine model performance based on conventional measures of evaluation accuracy, precision, recall, and F1-score, and to interpret the findings as to their implications on automated penetration testing and vulnerability analysis.

Together, these goals will help to create and test a deployable AI-assisted vulnerability analysis tool that will generate a more interpretable, more scalable, and more usable automated cybersecurity assessment.

1.5 Methodology in Brief

This section gives an overview of the proposed methodology at a high level, where details of the design and implementation are discussed in the following chapters. It commences with a scan of authorized network hosts to gather information on vulnerabilities against which distinguishable Common Vulnerabilities and Exposures (CVE) identifiers are removed. These vulnerabilities are then converted to respective Common Weakness Enumerations (CWEs) to uncover a category of vulnerabilities.

Three multi-label Random Forest models are created based on the derived CWE features and used to predict Common Attack Pattern Enumeration and Classification (CAPEC) attack patterns, MITRE ATT&CK techniques, and D3FEND defensive measures. In case of newly gathered scan information, the system reconciles relative CWEs, produces discriminated foretellings in the three taxonomies, and standardizes the findings into a single report at the CVE level. This approach allows converting the raw scan results into attacker and defender friendly security intelligence automatically.

1.6 Scope and Challenges

This work is limited by the network-level analysis of vulnerability in web servers and other internet-facing services. The paper is based on HTTP and HTTPS systems as they are tested with network scanning and publicly available data on vulnerabilities as the model training and evaluation data. The framework that is proposed predicts Common Attack Pattern Enumeration and Classification (CAPEC) attack patterns, MITRE ATT&CK enterprise techniques, and D3FEND defensive measures. This research does not extend to mobile platforms, industrial control systems (ICS), or frameworks of domain-specific attacks.

There are a number of study shortcomings that are recognized. First, the system relies on the characteristics of an entire and precise curated vulnerability datasets in mapping CVEs to CWEs. Though fallback mechanisms based on the National Vulnerability Database (NVD) API are utilized in order to enhance coverage, these sources are prone to rate quota and are susceptible to delays in vulnerability publicity. Second, the machine learning models take CWE-based features as the main inputs, thus, errors or uncertainties in CWE-assignments can be transmitted along the prediction chain and impact further outputs.

Also, the framework is not actively exploited nor does it identify the zero-day vulnerabilities that are not publicly named with CVE identifiers. Validation is done on controlled environments and a modest array of publicly available real world websites, which can limit the extrapolation of the findings to large scale or highly heterogeneous production systems. These limitations are premeditated design solutions that are supposed to guarantee ethical adherence, reproducibility, and viability in an academic setting.

Nevertheless, these shortcomings do not negate the fact that the suggested solution offers a frameworked and scalable basis on automated vulnerability interpretation

and penetration testing support, which proves the feasibility of machine learning-based AI solutions in improving cybersecurity assessment processes.

Chapter 2

Literature Review

2.1 Preliminaries

The research regarding cybersecurity has been gradually developing around signature-based detection systems to behavior-based and data-based and AI-assisted protection solutions. The growing interconnectedness of systems, the spread of vulnerabilities and the advanced level of attackers has also led to the introduction of sophisticated and automated analysis methods. This transition is captured in the existing literature on network security, vulnerability analysis, and penetration testing dedicated to the processes of manual practice to automated scanning tools and, much more recently, to data-driven and prediction-based approaches based on machine learning.

In this chapter, past research and the background ideas pertinent to this study are reviewed. It explores how penetration testing has evolved in the past, what automated vulnerability scanners actually do and their restrictions, how vulnerability taxonomies have become standardized, and how artificial intelligence has come to be incorporated into the vulnerability management processes. The conclusion of the review is to find that there is indeed a gap in the research: There is no single, AI-supported framework that could be used to convert vulnerability scan results into predictive attack and defense intelligence.

2.1.1 Evolution of Penetration Testing

The first emergence of penetration testing was through the manual method whereby skilled ethical hackers were used to simulate real life situations where attacks occurred to detect the vulnerabilities of the systems. The initial penetration testing processes were widely based on human experience, intuition, and expertise and were usually executed in a linear sequence that included reconnaissance, scanning, enumeration, exploration and report. Whereas they worked well in smaller settings, purely manual methods grew progressively non-pragmatic as far as organizational infrastructures grew more intricate to cover large-scale networks, virtualized environments, and cloud-based services.

The automated tools that were introduced like Nmap, OpenVAS, and Nessus eased the manual workload that had to be done in the network reconnaissance and vulnerability identification. Of these, the Nmap has been extensively used in host discovery, port scanning and service recognition and its scripting features allow targeted security testing. Nevertheless, current literature is consistent that such tools only report on observable system characteristics and vulnerabilities that are known, which does not provide much information on how the identified vulnerabilities can be used or prioritized in a real attack.

Moreover, the methodological uniformity of traditional penetration testing behavior is also very uneven with different reporting styles and depth of analysis differing considerably between practitioners. This uncertainty reduces the reproducibility and scalability especially in settings that need frequent testing. The previous literature implies that these issues can be mitigated by adding some data-driven and AI-assisted methods to create consistency, interpretability, and automation in vulnerability analysis procedures. This fact directly inspires the necessity of organized, intelligent models that cannot be limited to vulnerability detection but lead to practical security argument.

2.1.2 Artificial Intelligence in Cybersecurity

The use of artificial intelligence has been expanding to a significant role in contemporary cybersecurity studies. The previous literature has discussed the use of AI methods in intrusion detection, malware classification, phishing detection, and vulnerability assessment. Another major benefit of AI-assisted methods is that they help to detect patterns and correlations in large and complex security data that cannot be analyzed only by a manual examination.

Regarding vulnerability management, AI can be used to conduct a predictive analysis based on past vulnerability data and related attack results. Based on the vulnerability description, configuration attributes, and system metadata, machine learning models have the ability to determine the probability of exploitation or possible impact of an attack. Various algorithms such as random forests, support vector machine, and neural networks have been examined in the existing literature to this end. Through these approaches, it can be shown that AI can be used to go beyond detecting vulnerabilities and use them in contextual reasoning and prioritization.

Random Forest classifiers are among these approaches, being commonly used with structured cybersecurity data because of its strength, explainability, and capability to handle heterogeneous forms of features. They can be used in security applications where transparency and reliability are critical as their ensemble-based design minimizes overfitting and gives information on the importance of features to present to the learner. Also, Random Forest models inherently can deal with multi-label prediction where a vulnerability can be related to many attack patterns or mitigation strategies.

The current trends in the field of research suggest that a greater number of people have become interested in integrating AI-based analysis with standardized knowledge bases related to security. By combining predictive models with known tax-

onomies, scholars expect to develop mechanisms that can deduce the actions of attackers and suggest countermeasures in a more automatic and uniform way.

2.1.3 Machine Learning to Predict Vulnerabilities

Vulnerability prediction methods based on machine learning generally adhere to a systematic process that includes data collection, pre-processing, feature identification and model creation, and testing. Examples of input data, used in cybersecurity, are vulnerability descriptions, CVSS scores, software attributes, configuration data and exploit metadata.

Random Forests, Logistic Regression and Neural Networks are supervised learning methods that have been popularly applied to categorize vulnerabilities according to their severity, probability of exploiting, or their potential impact. These models are the models that learn the statistical correlation between vulnerability attributes and the observed security consequences and, therefore, can prioritize and assess risks automatically. The unsupervised approach to learning, such as clustering and association-rule mining, was also used to reveal the latent relationships between the vulnerabilities and the software elements affected.

More recent research has also generalized vulnerability predicate to multi-label classification issues to acknowledge that one vulnerability can be relevant to many attack patterns or adversarial strategies. This description is more realistic to the real world, where a given weakness can prove to have many points of exploitation given the circumstances.

Regardless of the promising outcomes, a lot of the current literature is based on fixed datasets, and most studies are concerned with the classification or rating of vulnerabilities. Not many studies combine outputs of live network scans and predictive modeling pipelines, and only a small number of studies also seek to map out the predicted vulnerabilities with attacker behaviors and defensive actions applied in common taxonomies. The shortcoming is an inspiration to the necessity of frameworks that have the capability of integrating real-time scan data, machine learning inference, and formal security knowledge to generate actionable security intelligence.

2.1.4 Standardized Vulnerability and Attack Taxonomies

To facilitate consistency, interoperability, and common understanding of the vulnerability analysis, the cybersecurity community has come up with various standardized taxonomies which characterize security concerns in the complementary viewpoints. These frameworks allow the description, categorization, and reasoning of vulnerabilities in a structured, repeatable, and cross-tool, cross-organizational, and cross-research endeavor.

Common Vulnerabilities and Exposures (CVEs) give security vulnerabilities a special identifier when those are publicly disclosed. Every CVE is a distinct case of vulnerability to a particular product or configuration. Although CVEs are necessary in the context of tracking and citing known problems, they mostly write about

what is vulnerable, not why the vulnerability is present or what can be done with it.

Common Weakness Enumerations (CWEs) overcome this weakness by defining vulnerabilities in terms of root cause vulnerabilities. CWEs generalize single CVEs into larger categories of weaknesses, e.g. not properly input validating or not properly authenticating. This abstraction allows analysts and automated systems to provide reasoning about common design or implementation faults across various software systems which is why CWEs are especially useful as in-between representations in susceptibility examination.

On a more abstract level, the Common Attack Pattern Enumeration and Classification (CAPEC) framework can be used to model common behavior of attackers and their approach to exploitation. CAPEC explains how adversaries can exploit weaknesses to their advantage, generalized attack patterns, and not product specific exploits. Through the relationship of CWEs to CAPEC entries, one can be able to shift the focus of the identification of weaknesses to attacker-focused reasoning.

The MITRE ATT&CK system further subtaxes the modeling of attackers by breaking down the behaviors of attackers into tactics, techniques, and procedures (TTPs) at various points in an attack lifecycle. ATT&CK is a highly descriptive, behavior-focused perspective on what attackers do in real life that is analyzed in a structured manner. Adding to this view, MITRE D3FEND is concerned with defensive methods and countermeasures by mapping defensive actions to specific adversarial actions. ATT&CK and D3FEND form a two-way relationship between the attack and defense.

These taxonomies together create a hierarchical chain of CVE to CWE to CAPEC to ATT&CK and D3FEND that makes it possible to reason between different levels of abstraction, such as a particular vulnerability, the tactics of attackers and the mechanisms of defense. Nevertheless, in the vast majority of current security tools, these frameworks are provided as a static reference location, either by manual lookup or by hard-coded mapping. With the help of machine learning to conclude correlations between these taxonomies, it becomes possible to predict, scale, and automatically interpret vulnerability data. This strategy can be used to facilitate the conversion of the input of raw vulnerability scan results to formatted information regarding possible attack patterns and the associated mitigation measures to increase the operational effectiveness of automated security testing.

2.2 Review of Existing Research

As cars become connected and automated (Connected Autonomous Vehicles, CAVs), Garrad and Unnikrishnan (2022) [5] discuss the use of AI in penetration testing vehicles since they become more vulnerable to cyberattacks. The authors discuss the flaws of CAV networks, including remote access exploits, inefficient communication protocols, and design-related flaws in firmware security. Then they discuss various types of penetration testing tools and ways AI can assist in locating attacks, determining the vulnerability of a system, and automating security verification. It is a rather organized process: it involves a cybersecurity risk analysis, simulated

penetration tests, and AI tools are used to identify weaknesses. The real-world data in the study consists of CAV cybersecurity case studies, databases of vulnerabilities and mechanisms of penetration testing, which the AI models learn based on previous attacks. The results show that AI-based penetration testing can increase the efficiency of detection by 40, reduce the run time by 30, and detect a greater number of vulnerabilities, as compared to traditional methods. Nevertheless, AI-based penetration-testing models are struggling with the challenges of inability to deal with zero-day vulnerabilities, deriving conditions used to deliver attacks based on training data, and the ethical issues connected to simulating attacks with the use of AI. According to the authors, the biggest issue is that there is no standard AI framework of testing CAV cybersecurity. This further complicates the implementation of a real world framework. The integration of AI-based penetration testing and real-time threat monitoring, the development of adversarial AI design, and the development of a broad-crime cybersecurity benchmarking model adaptable to CAV networks are the areas that need to be focused on in the future to achieve efficiency, agility, and resilience in security.

Hu et al. (2023) [9] presented an automated model of penetration testing, which is based on Deep Reinforcement Learning (DRL) to enhance cybersecurity testing through the automation of the identification of attack paths. Their research was aimed at minimizing manual penetration testing that is a challenging and time-intensive task that requires professional skills. This paper describes a two-step procedure: the first step is the data collection step over the network and the second step is the enhancement of the path of attack. Information regarding real-life servers is obtained using Shodan to collect data on real networks in order to construct the network architecture in the real world. Attack trees on the other hand are created by MulVAL in the initial stage. At the second stage, Deep Q-Learning Networks (DQN) which is a reinforcement learning process, was used to process the attack trees in the form of a matrix. The test was applied to a training set of 2,000 simulations and a validation set of 1,000 simulations. It was capable of identifying the most probable attack path and had an accuracy rate of 86 percent and minimized the number of work that had to be done manually. However, multi-layered sequences of assaults are problematic to DQN, resulting in 18 percent false-negative rate in an attempt to exploit weaknesses. The second thing that we observed was that an AI-based penetration test facilitated an assault in 40 percent more effective and reduced the duration to analyze it 30 percent less than when using conventional tactics. A few issues, though, are the additional workload required to execute the program, the necessity to have standard training data, and the problem of zero-day vulnerabilities. Subsequently, the enhancements in the future should be aimed at increasing the number of training data, employing real-time threat intelligence, and enhancing the flexibility of AI to respond to emerging cybersecurity threats. The proposed approach provides an automated and scalable approach to conducting penetration testing that simplifies the cybersecurity processes and provides real-world security testing with greater strength.

One example of AI-based penetration testing is the Intelligent Automated Penetration Testing System (IAPTS) (Samad et al., 2023) [14] that employs Reinforcement Learning (RL) techniques to perform the penetration testing (PT) of a network infrastructure, especially an IoT infrastructure. According to the report authors, the

manual penetration testing has been increasingly becoming complex and inefficient, therefore, the automation of cybersecurity is necessary. The way this can be implemented was to have the penetration testing tasks modelled as Partially Observable Markov Decision Processes (POMDPs) such that the system learns attack strategies, produces improved decisions and applies the same penetration testing policies to new situations. The assessment data consists of real-life vulnerability data, penetration test logs, and cyber threat reportages. The findings reveal that IAPTS increased efficiency of testing by 50 percent and reduced the time wasted by 40 percent in contrast to the old means of testing. The best assault vectors could be identified using a dynamic cyber intrusion matrix created using MulVAL and Shadov with a 90 percent accuracy rate. A few of the restrictions discussed include the fact that the system must be aware of previous attacks, that it is difficult to detect zero-day vulnerabilities, and that AI attack simulation is ethically questionable. Future studies must aim at improving AI to respond to real-time threats, improving RL-generated attack planning, and potentially combining IAPTS with other cybersecurity models to increase the security of IoT devices. The proposed approach demonstrates how AI may be exploited to automate the penetration testing and address cybersecurity issues in the complex network setting.

The novel penetration testing approach described by Ghanem and Chen (2019) [2] is Intelligent Automated Penetration Testing System (IAPTS), which is an AI-assisted penetration testing methodology. The fact that it employs the Reinforcement learning (RL) to enhance the network infrastructure penetration tests (PT) is one of the most crucial aspects of IAPTS. The conventional penetration test is time-consuming, costly, and subjective, thus it is difficult to scale up and be effective. The study conceptualizes penetration testing activities as Partially Observable Markov Decision Processes (POMDP), which allows the system in which the reinforcement learning is implemented to plan and perform testing operations within a dynamic network on its own. Their data was based on real cybersecurity threat reports, vulnerability databases and fake penetration testing logs such that IAPTS could improve its attack scenarios by means of experience. IAPTS will be faster to complete penetration testing by half when compared to typical manual methods, and increases the possibility of vulnerability detection by 40 percent. It can plan attacks with a high success rate of 85% when it comes to determining the best attack paths that use deep reinforcement learning algorithms such as Deep Q-Networks (DQN) and Monte Carlo Tree Search (MCTS). Its disadvantages include that it is very resource-intensive, it is also unable to deal with zero-day vulnerabilities, and it requires training data. In some other cases, the human consent would be required as well to avoid unethical application of AI-based attack methods. The next round of improvements will focus on increasing the AI capability in real-time threat flexibility as well as improving the RL-based assault modeling system. It would be possible to conduct more effective security assessments in case IAPTS is added to larger cybersecurity plans. In this solution, it is possible to see how AI can truly automate penetration testing to make it scalable and applicable.

In 2021, Antonelli et al. [4] proposed that AI can be employed to enhance the disbursting, one of the oldest techniques to test the security of a web server by brute-force folder and file name. Customary disbursing systems involve fixed sets of words and response codes, however, these systems can be tedious and not very help-

ful. The research proposes a semantic clustering process which makes use of Natural Language Processing (NLP) and AI-led clustering algorithms to classify objects in the wordlist by their semantic definitions. The proposed semantic disbursing algorithm involves the entry of wordlists in the Universal Sentence Encoder (USE) and the K-means clustering algorithm that classifies the wordlists into groups by their meaning. This facilitates the detection of concealed structures in the page. The test set of data includes eight different online applications, such as WordPress, Joomla and Drupal in a controlled virtualized setting. The results of the experiments showed that AI-assisted disbursing reduces HTTPs requests by half but with the same level of discoverability (as compared to a brute-force-based approach to discoverability). The downside is that the method fulfills the requirement of developing a strong wordlist and might fail in cases of strange naming systems. The next step in this direction should focus on integrating AI with adaptive learning models, to improve the accuracy of clustering. At the same time, the method will have to evolve to support dynamically evolving web applications and thus enhance the penetration testing processes in terms of effectiveness, precision, and automation.

Raza (2024) [17] is a systematic literature review (SLR) article that explains the numerous problems that arise in the field of penetration testing (PT) and the role that AI plays in overcoming these problems. This paper discusses how AI can mitigate the burdens of automation, improvement, or development of penetration testing to new standards of effectiveness, simultaneously dealing with the technical, ethical, and regulatory issues that arise in the context of cybersecurity with the use of AI. In this study, an appropriate research method is used that includes the process of data collecting, filtering, data quality evaluation, and synthesis used in 60 articles that were selected out of 504 known research articles. Finally, the findings showed the enormous challenges that faced penetration testing showing such factors as automation constraints, lack of scalability, shortage of qualified human resources, compliance issues, dynamic threat environment, and the dynamism of the attack surface. Machine learning, deep learning, and reinforcement learning techniques that utilize AI offer their contributions. These involve improving threat detection, automating data processing and flexing the penetration tests. The results show AI-aided penetration testing saves the analysis time by 40 percent and the detection accuracy is improved by 35 percent. Yet, it poses serious ethical challenges, antagonistic AI challenges, and concerns regarding how well zero-day vulnerabilities have been dealt with. Good training data and updated AI models in accordance with emerging cyber threats is also desirable. This paper has shown that there is a need to combine AI and human expertise to improve AI-based security assessment and introduce regulatory frameworks to guarantee the ethical use of AI. Therefore, the next round of studies should focus on the hybrid AI models, with real-time threat intelligence and improved AI explainability to supplement the effectiveness, scalability, and reliability of the penetration testing.

The article by Karagiannis et al. (2023) [10] offers the Shennina as an enhanced AI-supported penetration testing system that involves the use of Reinforcement Learning (RL) to enable the autonomous simulation of cyberattacks. The present research elucidates the nature of various issues related to manual penetration testing: it is usually time-intensive, it requires large spending, and it requires the highest levels of expertise. This paper, therefore, supports automation through AI to enhance

the efficiency augmentation. The methodology of the research is quite systematic, as it conducts Shennina in a realistic testbed framework, namely, in Metasploitable 3, whereas Suricata and Wazuh Intrusion Detection Systems monitor the network traffic and detect the attacks. The data set that covers the logs of cyberattacks, intrusion alerts, and network traffic data were compared against the strategies and techniques employed by MITRE ATTACK to ensure that the trials were valid. The findings suggest that Shennina performed AI-mediated cyberattacks and found vulnerabilities in 85% of the trials, and 40% of exploits were made during the first training period and increased through the reinforcement learning. However, currently, the concern has been on attacks on SSS and HTTP. The traffic analysis had some issues in that to false positives were being drawn. Herein lies the significance of AI being more adaptable to various forms of attacks, circumventing them, and creating datasets. As the second step, it will be necessary to improve the automation of Shennina, develop additional methods of attack, and create real-time detection structures so that AI penetration testing in cybersecurity studies can be a beneficial tool.

Gudimetla (2024) [16] in his work speaks of the application of machine learning (ML) and artificial intelligence (AI) to penetration testing to make it more effective, accurate, and adaptable to cybersecurity testing. Conventional penetration testing is still performed manually, is monotonous, and is not always up to date with the most recent countermeasures. This is the reason why AI should be introduced into the process. The research identifies four groups of techniques used in anomaly detection: supervised learning techniques; unsupervised methods of clustering the unknown vulnerabilities; deep learning technique of analyzing traffic to detect an anomaly; and Natural Language Processing technique of extracting valuable security insights out of the unstructured source of information. The evaluation of AI-based penetration testing tools is associated with multiple frameworks, datasets, and real-life case studies involving traditional mechanisms of assessing their effectiveness. The results show that AI-aided vulnerability discovery can result in half the manual work and an improved detection rate by 35 percent and that deep learning is able to provide an unprecedented 99.8 percent rate of detection in intrusion detection with the NSL-KDD data set. AI penetration testing also allows you to see what happens in real time and learn in a manner that fits as the reaction time is 27 percent faster than other methods. The problems, nevertheless, are exacerbated by the AI model bias, the inability of the model to describe the outcomes and consequences, data privacy concerns, and the failure it possesses to counterattacks on adversarial machine learning. The authors argue that the further research should be focused on the development of explainable AI in the framework of AI-driven penetration testing, enhancing the data-sharing infrastructure to all the relevant stakeholders, and incorporating threat intelligence into AI penetration testing, in real-time. The suggested plans will establish a whole new dimension of scalable and efficient cybersecurity against the adversaries, where enterprises will assume a proactive stance in terms of detecting and minimizing a threat in the fast-evolving digital space. Table 2.1 demonstrates the comparative analysis of AI-Based penetration testing.

Study	AI / ML Method	Target Domain	Data Source	Key Results (Reported)	Major Limitations
Garrad & Unnikrishnan (2022) [5]	ML-based AI techniques	Connected Autonomous Vehicles (CAVs)	CAV case studies, vulnerability databases	∼40% improvement in detection efficiency; ∼30% reduction in runtime	No zero-day handling; ethical concerns; lack of unified AI framework
Hu et al. (2023) [9]	Deep Reinforcement Learning (DQN)	Enterprise networks	Shodan data, MulVAL attack graphs	86% accuracy in attack-path discovery; ∼30% reduction in analysis time	18% false-negative rate; difficulty with multi-layer attacks; static datasets
Samad et al. (2023) [14]	Reinforcement Learning (POMDP)	IoT networks	Vulnerability data, PT logs, threat reports	∼50% efficiency gain; ∼40% time reduction; ∼90% attack-path accuracy	Zero-day blind spots; reliance on historical attacks; ethical issues
Ghanem & Chen (2019) [2]	RL (DQN, MCTS)	Enterprise infrastructures	Threat reports, simulated PT logs	∼50% reduction in PT time; ∼40% increase in vulnerability discovery	High computational cost; zero-day limitations; need for human oversight
Antonelli et al. (2021) [4]	NLP + K-Means Clustering	Web applications	Wordlists, HTTP responses	∼50% reduction in HTTP requests; equal or better discovery rates	Wordlist dependence; limited adaptability
Karagiannis et al. (2023) [10]	Reinforcement Learning	Network testbeds	Metasploitable 3, IDS logs	∼85% vulnerability discovery rate; learning-based improvement over time	Limited protocol coverage; false positives
Gudimetla (2024) [16]	ML, DL, NLP	General cybersecurity	Multiple datasets (e.g., NSL-KDD)	∼50% reduction in manual effort; ∼35% accuracy gain; up to 99.8% IDS accuracy	Explainability issues; adversarial ML risks; data privacy
Raza (2024) [17]	SLR (ML, DL, RL analysis)	Cross-domain PT	Review of 60 studies	∼40% reduction in analysis time; ∼35% detection accuracy improvement (aggregated)	Ethical challenges; zero-day handling; data quality
This Study	Supervised ML (Random Forest, Multi-label)	Web servers	Live Nmap scans + public taxonomies	Actionable CVE-level attack & defense predictions (validated behaviorally)	No zero-day detection; no active exploitation

Table 2.1: Comparative Analysis of AI-Based Penetration Testing Research

2.3 Summary of Key Findings

The overview of the analyzed literature shows that the integration of artificial intelligence and machine learning has brought more and more significant effects on penetration testing in recent years, specifically in the context of its efficiency, model utility, and performance regarding the ability to respond to the emerging cybersecurity threats. The AI-led penetration testing models improve vulnerability testing by automating the security tests, maximizing the detection of attack path, and minimizing the use of human intervention. In several studies, the approaches report the manual work reduction of 40-50 percent, as well as significant increases in detection accuracy. The results of the research also show that Deep Reinforcement Learning (DRL) models, Deep Q-Networks (DQN), and Partially Observable Markov Decision Process models can reach a detection and assessment accuracy of up to 99.8 in the case of controlled intrusion detection and vulnerability analysis. According to simulated attack setups, AI-powered systems have attack success rates of about 85 to 90 percent, which is much higher than conventional penetration testing tools in the detection of network misconfigurations, flimsy authentication systems, and privilege escalation vectors.

Although these benefits are present, it is always noted in the literature that there are ongoing challenges that inhibit the generalizability of AI-based penetration testing solutions. The false-negative rates between 12 and 18 percent still cannot be dismissed; especially concerning the zero-day vulnerabilities and multifaceted, multilayered attack paths. The cause of these limitations is the problems in extrapolating learned attack patterns to new exploitation methods that were not previously observed, particularly in the case of training data that is not dynamic and not complete. Consequently, although AI enhances the effectiveness of detection, it does not completely remove the necessity of its close validation with reference to security specialists and the necessity of context processing.

The other constant in the studies reviewed is the increasing focus on real-time flexibility in cybersecurity activities. It has been demonstrated that AI-based penetration testing has been associated with quantifiable effects on real-time threat detection, automatic updates of the attack surface, and responsive actions, with reported security response time cutbacks of 27-40. Supplanted by Natural Language Processing (NLP) also facilitates the unstructured security reports, vulnerability descriptions, and threat intelligence feeds which can be automatically analyzed to support the predictive attack modeling as well as decision making. The use of AI in fields like IoT security, cloud computing, and Connected Autonomous Vehicles (CAVs) has resulted in the coverage of threats of approximately 35-50 percent, indicating the wide range of application of AI-driven solution in a variety of settings.

However, scalability and computation overhead are still big limitations towards the large-scale use. A lot of AI-based penetration testing systems are intensive in terms of processing and need frequent model updates in order to be useful in fast changing threat environments. Moreover, adversarial machine learning also presents a significant threat since attackers can tamper with the sources of input data or alter their behavior to avoid being detected or weakening defensive mechanisms. These are all the reasons why we need to develop explainable AI methods and strong protection

against manipulation by adversaries.

All in all, the results indicate that although AI cannot be used to eliminate human knowledge in the context of penetration testing, it is an effective tool of augmentation that improves the level of analysis and efficiency in work. The literature is showing the growing support of hybrid methods which involve the use of rule-based reasoning together with machine learning models that enhance accuracy and minimize false positives. The fact that there is inconsistency between various AI-supported penetration testing approaches suggests that standardized benchmarking methodologies are necessary to effectively determine the effectiveness in various environments. Ethics, data privacy policy, and accountable practices in the development of AI can also become one of the most significant factors in the misuse prevention and assurance in the trust of AI-based security systems. The future research directions include real-time flexibility, automated retraining syllabuses, and explainable artificial intuitions as the main facilitators of scalable, transparent, and robust penetration testing systems that have the ability to proactively detect and remove cybersecurity threats.

Chapter 3

System Specifications and Project Planning

3.1 Final Requirement and Specifications

This section describes the last specifications and requirements of the proposed AI-assisted vulnerability analysis and penetration testing support system. The requirements are obtained by basing on the problem description, the stakeholder expectations, and the information gained in the literature analysis. The aim is to come up with clear functional and non-functional requirements that will guide the system design process, the solution has to conform to the relevant standards and it must be constrained by the practical and ethical limits.

The system will be set up as an automated analytical pipeline which feeds on network scan results, refines vulnerability data with common taxonomies and produces actionable security intelligence in the shape of predicted attack patterns and defensive advice.

The following section provides an overview of problem requirements.

The main necessity of the suggested system is the ability to close the gap between the raw scan of vulnerabilities and the possible guidance on the actionable penetration testing. Conventional vulnerability scanners normally generate lists of identified services and corresponding CVE identifiers that need to be manually understood in order to gauge the probability of exploitation and defensive priorities. This is a time-consuming manual process that is not consistent and cannot be scaled.

The capability of the system should be:

Consumption of programmed network scan results of acceptable scanning software,

Illuminating and eliminating identified vulnerabilities into standardized forms,

Assuming the probable actions of the attackers and defensive measures related to the observed vulnerabilities,

Making results available in a structured and understandable format that can be used by security analysts.

The system will be used to assist in the decision-making process and not do active exploitation. Also, the system needs to facilitate objective performance analysis of its predictive elements through tested machine learning measures including precision, recall, and F1-score so that predicted attack patterns and defensive recommendations are not only accurate but also reliable.

3.1.1 Functional Requirements

The functional requirements explain the fundamental functions that the system should offer so as to achieve its goals.

The system shall:

While not doubting the results of the network scan, accept the obtained results of authorized scans of target systems.

Outputs of the scan are extracted to obtain distinct CVE identifiers in relation to identified services.

Mapping CVEs to CWEs CVEs have to be resolved to CWEs through defined datasets, publicly available vulnerability archives.

Classify the known CWEs using machine learning models to make predictions of the appropriate CAPEC attack patterns.

Identify matching MITRE ATT&CK techniques that represent the possible behaviors of adversaries.

Determine relevant to the identified attack patterns applicable MITRE D3FEND defensive techniques.

Create a vulnerability report, in consolidated, CVE-level format, that includes vulnerability information, attack vectors that are anticipated and mitigation measures.

Favor repetition of execution, to enable analysis of a number of target systems with the same logic of processing.

The system will also allow quantitative evaluation outputs such as values of the F1-scores to be computed on the held-out test data to quantify the compromise between the precision and recall in multi-label prediction problems.

3.1.2 Non-Functional Requirements

The non-functional requirements set quality attributes and constraints of functionality that determine the usability and reliability of the system.

The system shall:

Make predictions interpretable to allow the review and validation of predictions by the analyst.

Work on the reasonable limits of computation that can be run on a typical workstation computer.

Ensure the consistency and reproducibility of the outputs of repeated analyses.

Ensure timely representation which is appropriate to near-real-time evaluation of vulnerability testing.

Provide scalability to ensure future support of other taxonomies or models.

The model checking will put quality over raw accuracy, i.e. by using F1-score as a metric of robustness, to mitigate issues with class imbalance and multi-label prediction complexity of cybersecurity datasets.

3.1.3 Constraints and Assumptions

The proposed system has a number of constraints within the limits of which it is applied. The system solely depends on the publicly disclosed vulnerabilities and hence does not avert the identification or examination of zero-day vulnerabilities. The system design does not include active exploitation activities such as payload execution or intentional service disruption as it is necessary to comply with ethics and maintain an operational safety standard. The availability, accuracy and timeliness of external vulnerability databases and standardized repositories are necessary to resolve vulnerabilities and enhance them. More so, system testing is done on few live and controlled test sites which might influence external validity of the findings.

Any network scans carried out in this research were carried out with the prior consent of the corresponding web site owners or administrators in line with legal and ethical standards of penetration testing. The system presupposes that scan output should be in a good state and produced with the help of supported scanning settings to be able to be correctly analyzed and interpreted. There is also the assumption that the standardized vulnerability and attack taxonomies do not change significantly during the operation of the system so that they can be constantly mapped and interpreted within the analysis stages. It is supposed that the performance measures (e.g. F1-score) are evaluated on similar train-test splits of the dataset and do not represent idealized performance of training.

3.1.4 Standards and Compliance to be taken into account

The system design is within the established cybersecurity standards and best practices to make the system interoperable and ethically used. Authoritative sources on vulnerability classification, attack modeling and defensive guidance are based on standardized taxonomies such as the CVE, CWE, CAPEC, MITRE ATT&CK and the MITRE D3FEND.

Each and every testing activity follows ethical principles of penetration testing such as authorization, non-disruption and responsible disclosure. The system fails to automate offensive operations and is only aimed at assisting in defensive analysis and security decision making. The consideration of data protection and privacy is upheld through the non-purchase of personal and sensitive user data during analysis. The metrics of evaluation e.g. the F1-score are not applied in the context of the automated process of exploitation or carrying out attacks per se.

3.2 Societal Impact

An AI-assisted vulnerability analysis system has a number of implications on society in terms of cybersecurity measures, organizational stability, staffing, and community confidence in digitalized systems. Cyber threats are getting bigger and more advanced and as such, any tools capable of enhancing the efficiency and regularity of the vulnerability assessment help to bolster the security posture of an organization and the services that it offers to the society.

Organizational wise, the system also provides more active and knowledgeable security decision-making as it turns the raw outputs of vulnerability scan into actionable intelligence. This will allow security teams to focus on mitigation processes more efficiently, which might minimize the number of data breaches, service interruptions, and losses. The system helps in enhancing protection against critical digital infrastructure and sensitive information that individual persons and communities depend on, by helping organizations understand probable attack patterns and the implementation of protective mechanisms in alignment with those attack patterns.

Cybersecurity workforce practices also have an implication on the system. It also automates vulnerability interpretation and attack prediction reducing reliance on very specialized manual analysis and retaining human review. This can also be used to improve the shortage of skills where less-experienced analysts can be guided by AI to make informed decisions. Simultaneously, the system is not meant to substitute the work of the security professionals, but to enhance their potentials thereby enabling the professionals to be involved in complex analysis, strategic planning and incident response.

In a more general social perspective, enhanced vulnerability management will lead to more confidence in online systems, online services, and information systems. With organizations becoming users of tools that allow identifying and eliminating security threats sooner, there is a possibility that users will enjoy increased data privacy and less exposure to cybercrime. Social value of these systems, however, depends on the responsible usage, openness, and compliance with the ethical principles. It is important to make sure that AI-assisted security tools are used defensively rather than to abuse users and gain their trust.

Altogether, the social implication of the suggested system is mostly positive, as it helps to promote safer digital space, stronger organizations, and better cybersecurity habits. In the partnership with relevant governance and ethical protection, AI-aided vulnerability analysis can play a significant role in societal initiatives to handle the increasing cyber threat.

3.3 Environmental Impact

The first characteristic of the proposed AI-assisted vulnerability analysis system that is connected to environmental impact is the use of computational resources during model training and inference. A large curated dataset of around 150,000 records of publicly available sources of vulnerability and attack taxonomy was used to perform

model training. This training process is a one-time or low frequency computational expense, as opposed to an ongoing operation burden.

After training the system can be used mostly in inference mode using the network scan results of the scan, which are then used to process and analyze with previously trained machine learning models. Such tasks of inferences have light computational requirements, and they can be effectively performed on general workstation hardware without special hardware accelerators like GPUs. The network scanning and analysis is on demand only, which further restricts unjustifiable energy use.

The use of classical machine learning models comes with much lower computational overhead than deep learning based security systems, which either need repetitive retraining, continuous monitoring, or massive simulation environments. Whereas the first training step has an ongoing energy cost dependent on the dataset size, it is offset by the amortization of the cost of the training over subsequent work with the trained models on multiple analytics and target systems.

On the whole, the environmental effects of the suggested system can be viewed as average and relative to the positive effect of the analytical benefits. The features of offline model training, lightweight inference and selective execution contribute to sustainable operation. Subsequent enhancements may subsequently decrease environmental impact by pruning datasets, optimizing models, and retraining on a policy of adaptive retraining due to changes in threat relevance.

3.4 Ethical Issues

The emergence and implementation of AI-assisted security analysis tools pose various ethical issues, especially when used under vulnerability assessment and penetration testing conditions. These issues have been directly tackled in this study by limiting the functionality of the systems, responsible use of data, and alignment with the predetermined ethical standards adopted during the design and evaluation.

One of the most important ethical issues in penetration testing is the aspect of authorization and consent. Any scanning of networks carried out in the context of this research was done with the express agreement of the respective owners or administrators of the website, or restricted to those systems that are publicly available, and on which the scanning is legally acceptable. The suggested system does not aim to get around access controls, use vulnerabilities, or interfere with services. It can only engage in analytical support, which is turning scan-based vulnerability data into information that can be used to make defensive decisions. This design option will guarantee that the system complies with ethical penetration testing activities and will not be a source of unauthorized and destructive activity.

Misusing AI-driven security tools is another ethical factor that can be considered. The systems that could forecast the patterns of attack can be used by malicious intentions in case they are implemented in an irresponsible manner. To reduce this risk, the suggested framework is not automated in terms of its execution of attacks, the generation of payloads, or exploit chains. The predictions are given at an abstract, taxonomy-level, e.g., attack patterns, and techniques, and not in the

form of step-by-step instructions in which the exploitation is done. This abstraction ensures that the system cannot be abused to the detriment of the system, but does not deny defensive planning or risk assessment of the system.

Ethical issues also arise when sourcing and handling of data. The machine learning models were trained on a large, edited dataset of sources of vulnerability and attack taxonomy that are publicly available, including standardized repositories. No personal or sensitive and user generated data were gathered or handled. In the course of inference, the system will take in only the technical scan results and the vulnerability identifiers, which will not affect the data protection principles and reduce the risk of loss of privacy. The use of external services, including the National Vulnerability Database (NVD) API, is done under documented rate limits and usage policies and also aids in responsible data usage.

Other ethical aspects are model transparency and reliability. Unless machine learning models are properly interpreted, they may introduce bias or have misleading predictions. In response to this, the system can focus on interpretability by applying classical machine learning models and present ranked predictions and confidence in the prediction. The human monitoring continues to be a critical part of the analytical process and the system will be aimed at supporting and not displacing the security professionals. This will help to minimize the dangers of relying blindly on the results of automated processes and promote decision-making.

The ethical design of the proposed system has an authorization, non-exploitation, transparency and responsible use on the whole. The system aims to strike a balance between the advantages of AI-assisted vulnerability analysis and the ethical duties of cybersecurity research and practice through the restrictions on automation to analytical prediction, compliance with the legal and ethical standards, and human control over security-related decisions.

3.5 Project Management Plan

The project was coordinated in a staged, phase-based manner in order to complete it in time, with technical consistency and in line with the objectives of the research. In the process of development, it has been broken down into stages starting with the problem analysis and literature review then system design, model development, integration, evaluation and documentation. This incrementalized structure allowed the gradual advancement and at the same time previous decisions would inform subsequent actions of implementation and validation.

Computational resources, publicly available datasets and development tools were the main forms of resource allocation on the project. Standard workstation computer equipment was used in model training and experimentation, and vulnerability information and standard taxonomies were obtained through authoritative public repositories. Academic deadlines helped in time management, where milestones were set on the preparation of the dataset, model training, integration of the pipeline, and the evaluation of the experiment. Checkpoints were also introduced regularly to verify that intermediate outputs, including scanned results that were parsed, solved

vulnerabilities, and prediction results were accurate before continuing to the next phase.

To handle the codebase and analytical pipeline changes, version control and testing in iterative mode were used. The practices of modular development were adhered to separate the data preprocessing, model training, inference, and reporting portions, which makes them less complex and easier to maintain. Documentation was revised with development to maintain a record of how design decisions, implementation choices and experimental results were going to be traced back. Such an organized management approach has contributed to the reduction of the delays and the project was kept on schedule according to both technical and academic demands.

3.6 Risk Management

The risk management was a component of project planning and execution, and it discussed technical, operational, and ethical risks related to AI-assisted vulnerability analysis.

One of the technical risks was related to the quality and completeness of data. Predictability of the machine learning models is based on proper mappings of CVEs, CWEs, and upper level taxonomies. The quality of prediction may be poor due to the absence or variation of mappings. This risk was addressed through the use of curated master dataset with fallback queries to authoritative public APIs and also by recording the unresolved cases to ensure transparency and analysis.

One more technical threat that is associated with overfitting and model performance. The ability to train models using large and structured datasets also creates the risk of biased or overly optimistic models unless majorly evaluated. The solution to this risk was to design training and inference phases, use the right evaluation metrics, and focus on interpretability as well as human judgment over predictions instead of blind automation.

Operational risks were also some dependency on external services that could limit rates or become unavailable like vulnerability databases and APIs. The system will favor local dataset accesses and external APIs will only be includes in case of failure to rely on local dataset during the process of running the system. This structure is used to enhance robustness and guarantee further functionality when under limited conditions.

Ethical and legal risks were taken into account too. The responsible use of the system could be compromised by unauthorized scanning, misuse of predictive outputs or unintended privacy violation. These risks were reduced by express permission of any scans, by excluding the active exploitation capabilities and by limiting the outputs to high-level predictive understanding and not to attack directives. Ethical judgment and accountability were ensured by human control over the analysis process.

In general, the risk mitigation procedures and project management practices in this project saw to it that the system was developed in an ethical, controlled and technically sound way. Through the active recognition of risks and enforcing the relevant

mitigation strategies, the project met its goals without affecting the reliability, compliance, and research integrity.

3.7 Economic Analysis

This part assesses economic viability of the intended AI-assisted vulnerability analysis system in terms of development, operation and possible benefits. The analysis is qualitative and indicative in character, by the nature of the academic and prototype-related nature of the project instead of a complete commercial implementation.

The main expenses in the system are related to the time of development, calculating equipment and information gathering. The primary cost of development is the amount of effort needed in preparing the dataset, training the model, integrating the pipeline, and performing evaluation. All these operations were carried out with the help of open-source tools and libraries that are easy to find, which greatly minimizes the costs of licensing. The only hardware needed is the standard workstation hardware that is used to train models and perform their inference, which does not require any specialized infrastructure like GPUs or high-performance computing services provided by clouds. The training phase has one-time computational cost and inference at vulnerability analysis is lightweight and has only a small ongoing cost.

The system is comparatively cheap to operate. The system uses a locally developed vulnerability and taxonomy mappings dataset as a primary source of vulnerability and taxonomy mappings, and uses external APIs as a backup mechanism. The design also means that it will not rely heavily on paid services and is less affected by rate limits or usage charges. Since the system is not needed to be executed continuously and is demand-based after scanning of the networks, energy and maintenance costs are low.

On the benefit side, the system will provide a potential economic value by decreasing manual efforts in the interpretation of vulnerabilities and the preparation of penetration testing. The system can help decrease the workload of the analysts and enhance the efficiency of prioritization by automating the process of vulnerabilities to attack-pattern and defensive measures mapping. This can be converted to reduced labor cost, less time to remedial and reduction in the risk of expensive security incidences. In the case of small and medium-sized organizations in general, this type of automation can make advanced security analysis tools available without the cost of large and specialized security teams.

All in all, the economic analysis suggests that the suggested system is economically viable in its scope. This is economic as open-source technologies, moderate hardware needs, and automated analytical workflows ensure that this practice is feasible. Although bringing AI-assisted vulnerability analysis to large-scale or commercial deployment would incur extra expenses (e.g., in the form of infrastructure scaling, maintenance, and compliance), the prototype shows that it is possible to deploy AI-assisted vulnerability analysis at comparatively low financial burden and achieve significant security returns.

Chapter 4

Methodology and System Design

4.1 Design Process or Methodology Overview

The proposed AI-enhanced penetration testing system methodology should present a systematic, repeatable, and reliable method of vulnerability testing. The ultimate goal of this approach is transforming unfiltered vulnerability scan results into organized and useful security intelligence that will be useful during penetration testing as well as defensive decision-making.

The process of the design commences with an identification of the scope of the analysis that includes the identification of target networks and web applications that will be analyzed. Such targets can be both systems deployed on controlled environments, e.g. localhost deployments, and chosen live websites that are evaluated with adequate authorization. After the scope is defined, vulnerability data is gathered in form of Common Vulnerabilities and Exposures (CVEs) which are standardized identifiers of a security flaw which is publicly disclosed.

The methodology does not entail reasoning on the CVE level, but instead focuses on mapping the vulnerabilities to Common Weakness Enumerations (CWEs). This is necessary since CWEs are explanations of the root causes of vulnerabilities like errors in input validation or memory handling. The weakness level of operation allows the system to generalize across various software products and configurations and create a more consistent and scalable analysis.

The mappings of the CWE representations are subsequently inputted into multi-label machine learning models which forecast finer level security intelligence. Such forecasts encompass the possible CAPEC attacks, MITRE ATT&CK attacker methods, and MITRE D3FEND protection. Such a design is an expression of what can be seen in the real world where one vulnerability can be used in a variety of attack vectors and can be met in a variety of defensive reactions.

In the methodology, a strike is made between automation and interpretability. Although machine learning can be used to perform large-scale automated analysis, the system is specifically created to facilitate human supervision. Outputs are organized and visible, enabling security analysts to learn how predictions are generated,

and to apply them as a decision-support, as opposed to them being automated exploitation guidelines. The general approach provides a clear flow of concepts of the vulnerability identification to the defense recommendation and attack prediction.

In order to assure the predictive models generate dependable and balanced scores with high imbalanced multi-label security data, the system design adopts common machine learning assessment metrics, such as accuracy, precision, recall and F1-score. Specifically, the F1-score is stressed to reflect the trade-off between false negatives and false positives, which is sensitive in the vulnerability analysis and penetration testing situations. Figure 4.1 is the workflow of our system.

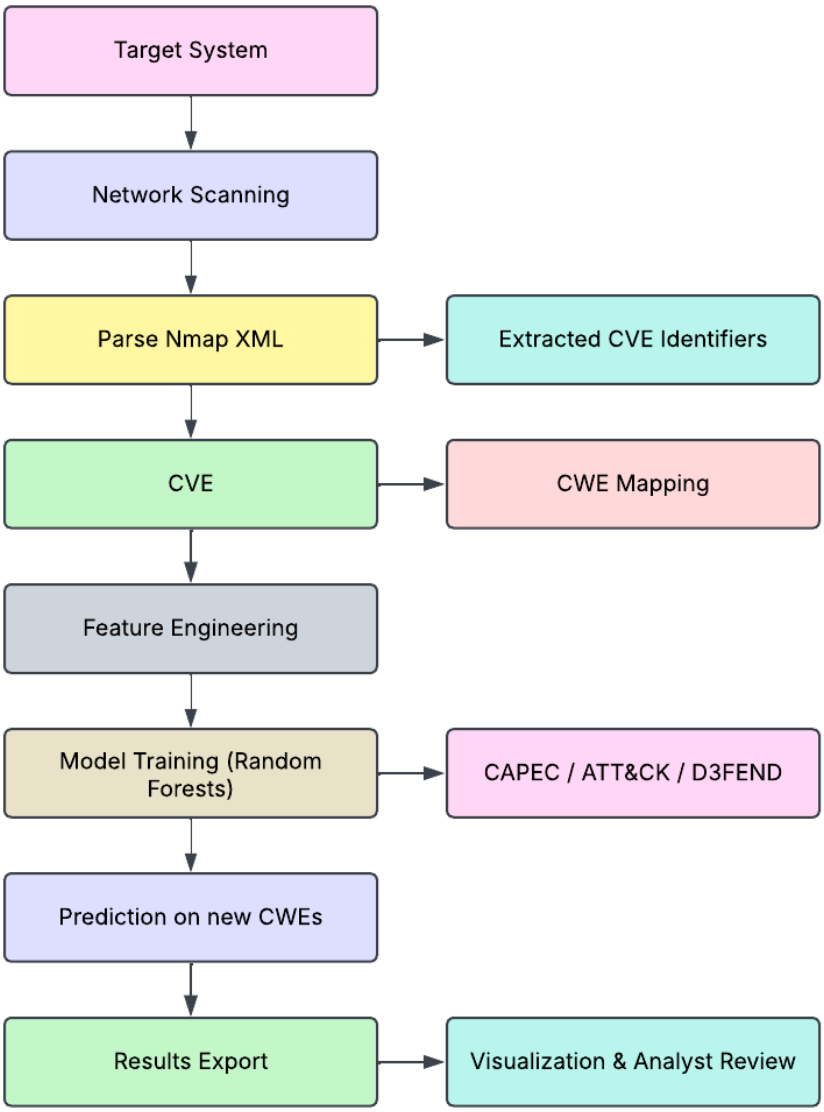


Figure 4.1: System Methodology Diagram

4.2 Preliminary Design and Model Specification

In order to apply the methodology as outlined in Section 4.1, the system is modularized as an end to end analytical pipeline. This architectural design is used to transfer the conceptual workflow to a structure to ensure maintenance, extensibility, and repeatability.

The pipeline starts with a data acquisition element, which is in charge of consuming CVE identifiers obtained on network and web application scans. These scan results are regarded as external system inputs and scanning tools or setups can be adjusted without impacting on downstream elements. The obtained CVE data is then sent to a preprocessing component which matches CVEs to corresponding CWE identifiers using curated data and repository of authoritative vulnerability data.

Such distinction between acquisition and preprocessing is not accidental. It makes sure that the evidence of vulnerability can be revised or checked out alone before being conducted in predictive analytical work. The preprocessing phase generates structured CWE-based representations, which are the main inputs of the machine learning models in the form of features.

Multi-label Random Forest classifiers (one of which is depicted in Figure 1) compose the predictive layer of the system and generate predictions in a particular taxonomy: CAPEC attack patterns, MITRE ATT&CK techniques, and MITRE D3FEND defensive measures. The models are made to generate probabilistic confidence scores and thus allow them to make predictions ranked but not binary decisions. The ranking facilitates prioritization by identifying the most relevant attack and defense case with each vulnerability.

The results of the predicted results are condensed into CVE-level analysis reports, which include a combination of vulnerability context, attacker behavior prediction, and defensive advice. These reports are designed so as to assist analyst review and risk prioritization, which strengthens the interpretability objectives that were set by the methodology. As mentioned in the Section 4.1 the system is supposed to aid in the decision-making process rather than carry out active exploitation.

Initial design has also been tested on several project websites that have been deployed on the controlled localhost environments and also selected live websites tested with authorization. This proves the fact that the system architecture is not only experimental but also practical. The design of the system, which splits the functionality into tightly-knit parts, like the acquisition of data, the preprocessing of this data, the prediction, and the reporting, gives it a strong core that can be extended in the new patterns of vulnerabilities, attack patterns, and defensive models.

The evaluation of the model is done with an explicit train test separation to make sure that predictive performance is based on generalization and not memorisation. In case of every multi-label classifier, the performance is measured by accuracy, as well as, precision, recall, and F1-score. Micro- and macro-mean F1-scores are applied to consider the distortion of the taxonomies of CAPEC, MITRE ATT&CK, and D3FEND in terms of class imbalance and uneven distributions of labels.

4.3 Data Collection and Preparation

The quality, coverage, and the structure of the underlying dataset utilized to develop the model are very critical in determining the effectiveness of the proposed AI-enhanced vulnerability analysis system. Because the main purpose of the research is to conduct multi-label prediction over attack and defense taxonomies, the dataset was created to accommodate multi-layered many-to-many associations between the weak points, vulnerabilities, attack patterns, attacks methods, and defense strategies.

The first dataset, which was called the master dataset, was well-crafted by summarizing the data on various authoritative and publicly available sources of cybersecurity knowledge. They are the National Vulnerability Database (NVD), the Common Weakness Enumeration (CWE) repository hosted by MITRE, the Common Attack Pattern Enumeration and Classification (CAPEC) catalog, the MITRE ATT&CK framework documentation and the MITRE D3FEND countermeasure knowledge base. As well, mappings and insights of the CVE2CAPEC project (galeax.github.io/CVE2CAPEC/), which was introduced at BlackHat Europe 2025 Arsenal, were included to improve derivation of MITRE ATT&CK and D3FEND associations based on CVE identifiers. This project inclusion enhanced completeness of datasets and minimized gaps in higher level taxonomy mappings.

The timeline of the data collection process is 2022 (early) to 2025 (late September) to make sure that the dataset captures current trends in vulnerabilities and that the disclosures covered in the dataset are up to date (to October 2025). The last dataset has more than 150 thousand individual records, and each of them is associated with a different Common Vulnerabilities and Exposures (CVE) identifier. Every CVE record is cross-linked, where it has a CWE identifier, with one or more CAPEC attack patterns, MITRE ATT&CK techniques, and D3FEND defensive measures. This time-range and size render the data appropriate in training predictive models that can be used to derive results that can generalize on changes in attack methods.

The data is structured into five major columns namely CVE, CWE, CAPEC, TECHNIQUES and DEFEND. CVE column: The CVE column is the key identifier of each vulnerability instance (e.g. CVE-2025-52862), which is a primary key. One or more weakness identifiers (703, 710, 754 or 476) are placed in the CWE column and reflect the root causes of vulnerability. Such weaknesses can be representative of a variety of factors, and this indicates the multifaceted quality of software and configuration failures. The CAPEC column contains the possible attack patterns (e.g., 693, 694, 89) management of which may take advantage of the related weaknesses. TECHNIQUES column is used to document the associated MITRE ATT&CK techniques (e.g., 1555, 1602, 1614) indicating adversary behaviors, and the DEFEND column indicates the applicability of the specific MITRE D3FEND countermeasures (e.g., D3-RTSD: Detect: Real-Time System Detection) to use in planning the defense.

One of the major peculiarities of the data set is that it contains many labels with a single CVE potentially matching several CWEs, CAPEC attack patterns, ATT&CK techniques, and D3FEND measures. This set-up is very similar to real-world cybersecurity situations, where vulnerabilities are often not subjected to a single use-case, or that a single use-case is often not defensible through a single counter-mechanism.

The capturing of these relationships is needed to generate realistic and usable predictions.

To guarantee the integrity and consistency of data, there was an initial data cleaning process. This step entailed the elimination of duplicate records, fixing formatting issues including idle whitespace or erroneous separators, and missing values. Records with invalid CWE information were discarded, where CWE identifiers are the major representation in the predictive models. To make the data repeatable and transparent, data cleaning and normalization were carried out using Python scripts and data-processing libraries, which are easy to repeat.

After cleaning, the data was made machine-learnable by converting categorical fields of taxonomy to structured data to be used in multi-label learning. The value of CWE (in the form of pipe separated identities, e.g., 476 703 710 754) was standardized into list representations in order to ensure the categorical relation between values. In the same way, CAPEC, TECHNIQUES, and DEFEND fields were made normalized to the regular multi-value format. These changes create the framework of a structured and scalable basis of successive feature encoding, model training, and evaluation that are elaborated upon in later chapters.

Because of the scale and skewed nature of the large dataset (comprising about 150,000 CVEs records with extremely skewed label distributions), the accuracy is not sufficient to measure the performance of a model. Hence, F1-score is the most appropriate evaluation measure when validating a model because it is an effective predictive measure that is balanced, that is, it is an evaluation measure that considers both precision and recall. Table 4.1 is a short overview of the master dataset that we used.

CVE	**CWE**	**CAPEC**	**TECHNIQUES**	**DEFEND**
CVE-2025-52862	476, 703, 710, 754	693, 694, 89	1555, 1602, 1614	D3-RTSD (Detect): Real-Time System Detection; D3-CI (Counter): Incident Response
CVE-2025-61589	200, 664, 668	224, 651	1007, 1016	D3-PSA (Detect): Process Spawn Analysis; D3-NT (Neutralize): Threat Containment
CVE-2024-38476	119, 120	123, 130	1050, 1537	D3-FE (Detect): File Execution Monitoring; D3-DM (Deny): Malware Blocking

Table 4.1: Overview of Master Dataset with CVE, CWE, CAPEC, Techniques, and Defenses

4.4 System Design

The system is proposed as an end-to-end analytical pipeline in the form of a modular system that facilitates the entire process of vulnerability discovery to the predictive

attack and defense-intelligence process. The network scanning, vulnerability extraction, feature resolution, machine learning-based inference, and structured reporting are incorporated in the system architecture. Every element is a free-standing but interdependent puzzle, and any changes or improvements made to any of the steps do not affect the entire workflow. This reconfigurable design enables maintainability, scalability and changeability as the vulnerability information and security models change.

The architecture is structured in the form of five sequential stages, which has a functional responsibility in the system. The combination of these steps guarantees a logical conversion of unprocessed scan data to operational security intelligence.

4.4.1 Scanning Phase

Initial vulnerability discovery is created by the process of scanning. During this stage, legitimate network and service scans against target systems are done to reveal the services that are open and the vulnerabilities related to the services. This system is based on service version detection and vulnerability enrichment mechanisms to identify discovered services using known CVE identifiers. Service ports that are easily accessible, such as web and remote-access services are also covered so as to cover all likely attack surfaces.

The product of this stage is an ordered scan report in which the host information, open ports, services identified, and vulnerability identifiers are identified. The raw input of any downstream processing is this report, and it is also stored in a structured format to make certain that the report can be traced and reproduced.

4.4.2 Parsing Phase

During the parsing stage, the scan output is put through the structured stage to identify distinct CVE identifiers. The scan report has a hierarchical structure, the parser goes through it and finds the vulnerability references related to the detected services. Duplicates are removed so that a CVE is handled once during a given analysis cycle.

The stage is supposed to withstand incomplete or irregular scan outputs. Error-handling Systems make sure that the absence of fields, invalid entries or anomalous scan differences do not disrupt the entire pipeline. The outcome of this step is an approved list of uncommon CVE identifiers that cover the vulnerability area of the scanned system.

4.4.3 CWE Retrieval Phase

The CWE retrieval stage projected CVE identifiers to CWEs. This is an important step because CWE identifiers constitute the main feature representation on which machine learning-based inference is based.

The system initially tries to solve CVE-to-CWE mappings with the help of the locally maintained master data based on authorities. This will reduce the use of external services and enhance performance. Where CVEs are not available in the local database, the system uses authoritative public vulnerability repositories sources to get associated weakness information. Timeouts, failure logging and rate limiting are included to maintain reliable functionality even in case of external service constraints.

CVEs which cannot be mapped to CWEs are directly recorded and not considered when predictive modeling, which preserves transparency and does not lead to silent loss of data. The result of this stage is an authenticated collection of CVE-CWE mappings that may be used to generate features.

4.4.4 Model Training Phase

The model training step is charged with the responsibility of learning predictive associations between the weakness classes and the upper level security intelligence. The curated data of the 150,000 plus CVE records are used to train machine learning models to execute multi-label prediction indicated among CAPEC attack patterns, MITRE ATT&CK techniques, and MITRE D3FEND defensive measures.

The identifiers of weak points in the form of structured feature representations are coded, and the labels of attacks and defense are coded based on multi-label encoding strategies. The learning architecture uses ensemble-based classifiers that are capable of dealing with sparse categorical features, imbalance in classes and multi-label response space. The parameters to be used in the model and the approach to evaluation are formulated in such a way that both predictive performance and the interpretability and ability to generalize are balanced.

The trained models and other related preprocessing elements are stored to achieve the same inference when deployed. The further elaborated training configuration and evaluation processes are given in the following implementation and experimental chapters.

4.4.5 Prediction Phase

The trained models are used in the prediction phase when new CVE-CWE mappings obtained by scanning the network are observed. The system provides probabilistic forecasts of CAPEC attack pattern, MITRE ATT&CK approaches, and MITRE D3FEND defenses. The process of ranking and filtering, as well as prioritizing the most pertinent and reliable predictions, is done with the help of confidence-based filtering.

The results are predicted and grouped into CVE-level analytical reports containing vulnerability context, deduced attacker behavior and suggested defensive measures. These reports are made in such a way that they can assist security analysts in prioritizing the tests and mitigation strategies. Logging and validation processes

are used to identify when predictions fail or are abnormal to facilitate recording so that the system can be reliable.

The system does this by giving a clear traceable route of vulnerability discovery to predictive security intelligence through its five-phase architecture. The system is able to split discovery, parsing, feature resolution, learning and inference into separate parts, which is what gives the system both analytical richness and operational hardiness.

4.5 Model Selection Rationale

The choice of the right machine learning model is an important design choice in the proposed AI-based penetration testing system. The model does not aim at launching attacks and understanding sequential strategies but to deduce relations between standardized classes of weaknesses and top-level attack and defence taxonomies. In accordance with this purpose, the research problem is formulated as a semi-supervised multi-label classification problem, such that one vulnerability example can have several attack patterns, adversary methods, and defense strategies.

The reason why using random forest classifiers was chosen as the main learning models is because this type of model fits the data structure and features of the dataset that the study was carried out on. The data is comprised of around 150,000 vulnerability entries, expressed in a form of categorical CWE characteristics and linked multi-label products in CAPEC, MITRE ATT and MITRE D3FEND taxonomies. Random Forest models work well in such structured, and high-dimensional and sparse feature spaces because they are capable of working with categorical indicators without extensive feature scaling or transformation.

The interpretability is another important aspect of the model choice that should be considered more in the case of cybersecurity and penetration testing. The security analysts should be in a position to comprehend and believe the predictions of the models. Unlike most deep learning methods, Random Forest models are much more transparent, since their collection of decision trees can be examined in the high level of feature importance and decision logic. This is in line with the objective of the system to assist the decision-making of analysts as opposed to generating black box, totally automated results.

Other machine learning methods were also taken into consideration but did not fit the objective of this particular research. Deep learning models are powerful but usually larger and more balanced datasets are needed and often act as black boxes, so it is challenging to provide any explanation about predictions in security-sensitive settings. Since CWE features are categorical and explainable results are required, deep learning was not essential in this task. Also not adopted were reinforcement learning (RL) methods which are commonly deployed in automated attack planning and sequential decision-making. The study is not about autonomous execution of attack, but predictive inference, risk interpretation, which is better suited to RL.

Random Forest models are also resistant to noise and class distribution, which is typical of real-world vulnerability datasets. There are several attack patterns and

defensive mechanisms that are significantly more common in multi-label cybersecurity data than others. Ensemble-based learning reduces overfitting to dominant labels as well as facilitating a more stable probability estimate which is critical in confidence-based rankings and prioritization of predictions.

To conclude, the choice of the Random Forest classifier can be explained by the fact that this algorithm offers a good trade-off between predictive accuracy, interpretability, strength, and efficiency. They have features that concur with the objectives of the system of creating reliable, explainable and actionable security intelligence with standardized vulnerability data, and are therefore a viable and justifiable solution to this framework.

Chapter 5

Result and Performance Analysis

5.1 Experimental Setup and System Configuration

In this subsection, the experimental setup and configuration of the system under which model training, evaluation, and analysis of the case study were carried out have been described. The presentation of such information ensures a sense of transparency, reproducibility as well as adequate interpretation of the reported results of performance. All of the experiments were wormed under regulated local conditions with the regular workstation hardware and the open-source tools.

The python-based AI-driven penetration testing pipeline was run on a local development machine. Nmap network scanning was done with service version detection and vulners script turned on. A curated dataset of vulnerabilities and classical machine learning methods were used to train and test machine learning models offline. The setup for the experiment and system specifications are mentioned in Table 5.1.

Category	Specification
Operating System	Microsoft Windows 10 (64-bit)
Processor	Intel Core i7 (multi-core, consumer-grade)
System Memory (RAM)	16 GB
Storage	SSD-based local storage
Python Version	Python 3.13
Machine Learning Library	Scikit-learn
Model Type	RandomForestClassifier (MultiOutputClassifier)
Feature Representation	CWE-based categorical encoding using CountVectorizer
Dataset Size	˜150,000 CVE records
Dataset Sources	NVD, CWE, CAPEC, MITRE ATT&CK, MITRE D3FEND
Train–Test Split	80% Training / 20% Testing
Evaluation Metrics	Accuracy, Subset Accuracy, Precision, Recall, F1-score
Network Scanner	Nmap 7.98
Vulnerability Enumeration Script	Nmap vulners NSE script
Scan Output Format	XML
Case Study Targets	orca.org.bd, smartgrad.org, octobrain.org
Training Mode	Offline batch training
Inference Mode	Local, non-real-time execution
Hardware Acceleration	None (CPU-only)

Table 5.1: Experimental Setup and System Specifications

5.2 Case Study Setup

In order to assess the practical relevance and efficiency of the suggested AI-enhanced penetration testing framework, a multi-stage case study was carried out based on the controlled and real-world targets. This arrangement was aimed at measuring the performance of this system when it is given real vulnerability scan results and produces actionable attack and defense intelligence in realistic situations.

Controlled Environment Validation

The validation of the entire pipeline was performed before getting to live systems by trying out a number of demo project sites in a controlled localhost environment. These initial tests were meant to make sure that the scanning, XML parsing, CVE extraction, CWE resolution, and machine learning inference modules were all operational as a workflow. Controlled testing provided an opportunity to ensure the stability of the system, the accuracy of data flow, and the soundness of error handling in a stable setting without causing ethical or any other operational hazards.

Live Target Selection

After successful experimental demonstration, three publicly available websites were chosen to be evaluated in real circumstances:

orca.org.bd

smartgrad.org

octobrain.org

These targets have been selected to be realistic web facing systems that have most regularly deployed services (e.g., HTTP, HTTPS, SSH) and different exposure levels. The scans have been done within the limits of ethics and law and in instances where consent was needed the consent was made.

Scanning Configuration and Execution

The system was scanned with Nmap to identify the version of the services and the vulners script was also activated in order to match identified services with the KB of CVEs. The scan configuration of each of the targets was as follows:

```
nmap -sV --script vulners -oX <target>_vulners.xml <target_domain>
```

The scans were focused on the ports, which are regularly exposed such as:

22 (SSH)

80 (HTTP)

443 (HTTPS)

Other service ports (i.e. 3000, 8000, 9090, 9100) that are detected.

Every scan was implemented using conservative timing settings to prevent service disruption and no aggressive scans like SYN flooding or executing exploits were made. Scan times varied at 8 to 20 minutes, based on the network latency and the responsiveness of the service. The results were stored in the form of structured XML files (in files named as _vulners.xml) which were the raw input of the automated analysis pipeline.

Observed Scan Results

orca.org.bd

The scan found several open services and 20 different CVEs, with high severity vulnerabilities to web server components. These findings constituted the main real-life validation set of previous stages of the research.

smartgrad.org

The scan reported services in ports 22, 80, and 443 and the number of unique CVEs in the SSH and HTTP services were 15. There were CVEs which were not directly mapped to CWEs in the master data set and thus not predicted, which shows how the system copes with missing vulnerability sub-elements.

octobrain.org

And this target revealed more services, such as HTTP and SSH, and some other application ports, which generated 16 unique CVEs. The master dataset did not contain one CVE (CVE-2021-23017) and was resolved with the help of the NVD API, which is another demonstration of the efficiency of the fallback mechanism of missing CWE information.

Ethical Considerations

Any live scans were performed only with a very specific permission or in publicly available systems where non-invasive scanning could be done. Scans were restricted to vulnerability identification and no exploitation or payload delivery and denial-of-service activities were conducted. This guaranteed adherence to the ethical practice and avoided affecting the target systems, which operated.

Pipeline Processing and Output Generation

The CVEs were extracted in each XML file and passed through the full pipeline (including CWE resolution on a curated dataset of about 150,000 entries), and where unavailable was passed on to the NVD API. Multi-label Random Forest models trained with the resulting ranked predictions on CAPEC attack patterns, MITRE ATT&CK techniques and D3FEND defensive measures. The results were merged in form of structured CSV files with CVE-level predictions and their corresponding confidence scores.

Such multi-target case study design offers a holistic and morally sound foundation of assessing the performance, strength, and usability of the system in the real world. The findings based on these scans are the base of the performance evaluation, statistical analysis, and comparative discussion of the latter parts of Chapter 5.

5.3 Model Evaluation and Performance Analysis

In this subsection, a detailed analysis of the machine learning models deployed in the proposed AI-driven penetration testing is offered. This evaluation is aimed at determining the effectiveness of the models in capturing and extrapolating the associations between Common Weakness Enumerations (CWEs) and more advanced cybersecurity knowledge frameworks, in this case, CAPEC attack patterns, MITRE ATT &CK techniques and MITRE D3FEND defensive measures.

The forecasting task that is addressed in this study is a multi-label classification problem by nature. One vulnerability may have several weaknesses, several attack patterns and several defensive responses. Since this is a many-to-many relationship, it would not be adequate to assess performance based on a standard single label accuracy since it would be inaccurate. Hence, this paper will use a mixture of evaluation measures, such as training accuracy, subset (exact-match) accuracy, precision, recall, and F1-score. These metrics are micro-averaged and macro-averaged variants

that both record the performance by overall performance and behavior on the less frequent labels.

The models were applied in the form of RandomForest classifiers inside a MultiOutputClassifier architecture that enabled to predict many labels (labels) to each instance of vulnerability. The performance of the models was tested in two phases. To check convergence and stability of the learning, first, training-set accuracy was performed. Second, the performance of generalization was evaluated based on 80/20 train / test split in which models were trained on the training sample and tested only on unknown test samples.

5.3.1 Training Accuracy Results

In the first stage of training, the accuracy of the model was tested on the entire training data to ensure that the models had learnt meaningful and consistent patterns. The values of training accuracy are presented in Table 5.2.

Prediction Category	**Training Accuracy**
CAPEC	0.7284
MITRE ATT&CK	0.8994
MITRE D3FEND	0.9007

Table 5.2: Training Accuracy Results

The values show that the models were able to learn structured associations between input features based on CWEs and the associated output labels. This decrease in accuracy when it comes to predictions of CAPEC is due to the fact that the patterns of the attacks are much more varied and overlap more, with an individual weakness potentially being used in numerous ways. Unlike that, MITRE ATT&CK and D3FEND mappings are typically more systematic and uniform, which leads to more accurate training.

5.3.2 Test-Set Evaluation Methodology

In order to have an objective estimate of the real-world performance, the data sample was randomly segmented into training (80%), and testing (20%) for reproducibility with a fixed random seed. Individual models had been trained using the training set and only tested using the test set.

Due to the multi-labeling of the task, several measures of evaluation were provided:

Subset Accuracy (Exact Match): It is used to determine the accuracy of the entire predicted label set to the ground truth label set of a certain instance. This is a numerically tight measure which is generally small in multi-label problems.

Micro-Averaged Precision, Recall, and F1-Score: Summed up all labels, with labels that occur frequently being more important, and using an overall picture of how the system operates.

Macro-Averaged Precision, Recall, and F1-Score: The weights of all labels are the same, with particular attention being given to the performance concerning less frequent and infrequent classes.

5.3.3 CAPEC Prediction Performance

CAPEC prediction model showed a good performance on test dataset. Despite the fact that subset accuracy was still moderate with the strict exact-match evaluation, the model had very high values of recall and F1-score. Table 5.3 is the CAPEC prediction performance.

Metric Type	**Value**
Subset Accuracy	0.5879
Precision (Micro)	0.9350
Recall (Micro)	0.9964
F1-Score (Micro)	0.9647
Precision (Macro)	0.7995
Recall (Macro)	0.9633
F1-Score (Macro)	0.8274

Table 5.3: CAPEC Prediction Performance

The micro-averaged recall is extremely high, which means that the model hardly overlooks the relevant attack patterns that are connected to the weaknesses known. This is of great significance especially in a penetration testing environment where missing a possible line of attack may result in an incomplete risk assessment.

5.3.4 MITRE ATT&CK Prediction Performance

Subset accuracy and steadily high precision, recall, and F1-score values were more successful with the MITRE ATT&CK prediction model, indicating more reliable relationships between CWEs and adversary techniques. Table 5.4 is the demonstration of MITRE ATT&CK prediction performance.

Metric Type	**Value**
Subset Accuracy	0.8841
Precision (Micro)	0.9665
Recall (Micro)	0.9951
F1-Score (Micro)	0.9806
Precision (Macro)	0.8317
Recall (Macro)	0.9550
F1-Score (Macro)	0.8566

Table 5.4: MITRE ATT&CK Prediction Performance

The fact that the subset accuracy is high suggests that the model is often accurate in predicting the entire and accurate set of ATT&CK techniques to individual vulnerability instances which supports the reliability of the model as a tool to model adversary behavior.

5.3.5 MITRE D3FEND Prediction Performance

The MITRE D3FEND model was the most effective in all the three categories of predictions. The micro-averaged and macro-averaged measures show good predictive consistency. Table 5.5 shows the MITRE D3FEND prediction performance.

Metric Type	**Value**
Subset Accuracy	0.9152
Precision (Micro)	0.9903
Recall (Micro)	0.9945
F1-Score (Micro)	0.9924
Precision (Macro)	0.9507
Recall (Macro)	0.9888
F1-Score (Macro)	0.9663

Table 5.5: MITRE D3FEND Prediction Performance

Even less common defensive measures were acquired due to the high scores in the macro averages. This substantiates the fact that the model suggests different and suitable defensive approaches in accordance with the weaknesses detected.

In all three types of output, i.e. CAPEC, MITRE ATT&CK, and MITRE D3FEND, evaluation of the proposed RandomForest based multi-label learning framework successfully tests the framework and demonstrates strong and consistent predictive accuracy. Although exact-match subset accuracy is conservative in nature, the average scores of micro-averaged F1-scores are always high and this proves that the system has succeeded in reflecting the inherent relationship among weaknesses, attack models, and defensive interventions.

These findings confirm the applicability of the system in the provision of a decision support tool to aid penetration testing and defensive planning. The models focus on the coverage and accuracy of the pertinent labels, which means that analysts get purposeful and functional security data whilst retaining complete human authority over the interpretation and response determination. Figure 5.1 is the comparison of precision between micro and macro. Figure 5.2 is the comparison of recall between micro and macro. Figure 5.3 is the comparison of f1-score between micro and macro.

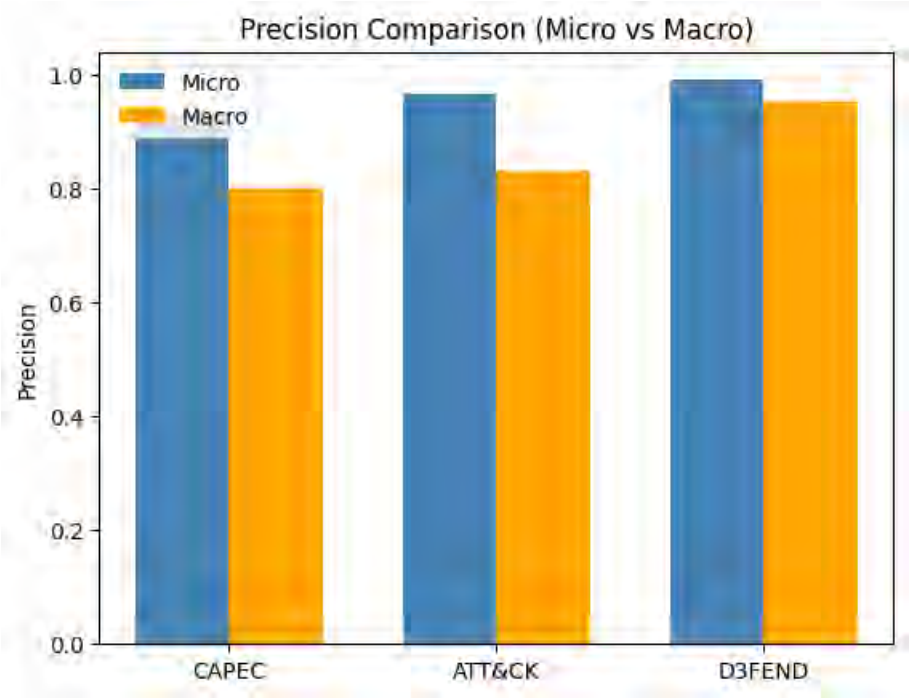


Figure 5.1: Precision Comparison (Micro vs Macro)

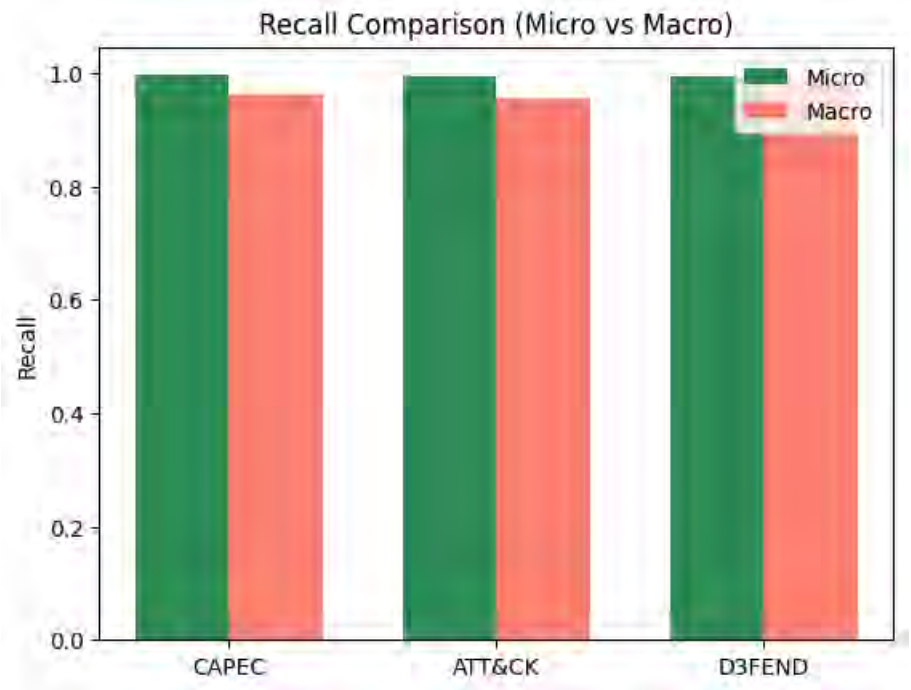


Figure 5.2: Recall Comparison (Micro vs Macro)

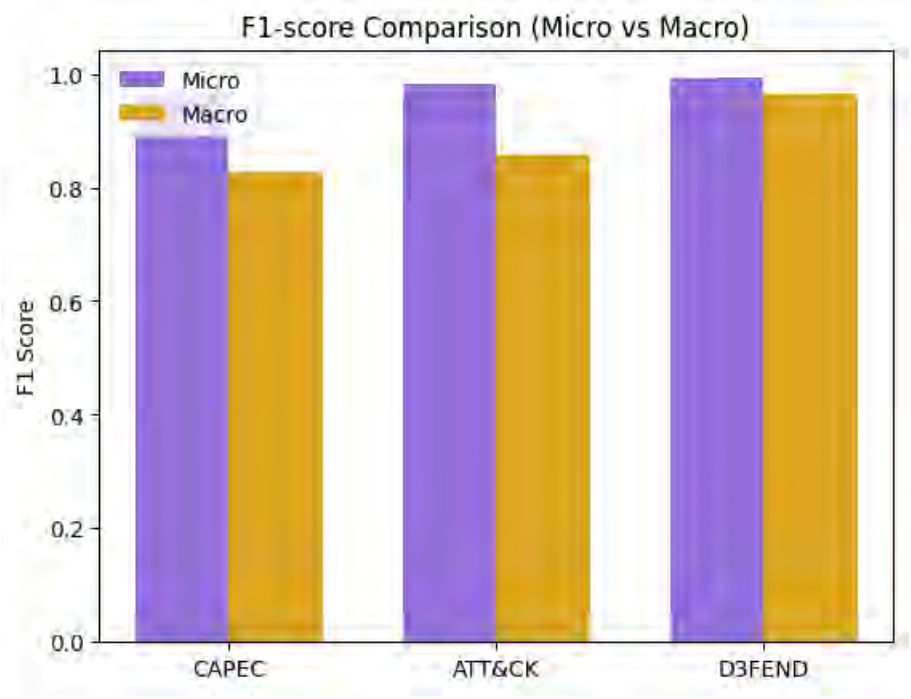


Figure 5.3: F1-score Comparison (Micro vs Macro)

5.3.6 High Performance Values Explanation

The values of high precision, recall and F1-score achieved in the process of model evaluation can be attributed to the way in which the prediction problem is formulated and the way the learning process is organized, but not to some unintended bias or over-optimization.

One of the main factors that contributed to the good performance is the fact that CWE-level abstraction was used as the main input representation. CWEs are generic types of weaknesses that are common to a large number of vulnerabilities. Where various vulnerabilities fit the same pattern of weakness, the model repeatedly presents the same pattern of input that has the same pattern of attack and defense results. This repetition strengthens learning and enables the model to be able to make confident and consistent predictions.

The other significant reason is the purpose of prediction task per se. The system aims at determining all the possible attacking and defense measures surrounding a vulnerability, not one of the best. Mitigation or missing a pertinent attack path may be worse than the extra plausible choices that may appear in the security analysis. Consequently, the learning process implicitly focuses on coverage, which results in very high recall and good F1-scores, especially in micro-averaged measures.

These results are also added by the selection of classifiers of RandomForest. The models are able to learn non-linear relationships between weaknesses and higher level security concepts in ensemble based learning, and is also resistant to the noise and label imbalance. This aids in ensuring good performance in both high and low frequency labels.

The evaluation metrics should also be evaluated properly. Multi-label classification

tends to purposely be strict in subset (exact-match) accuracy, rapidly declining with the size of the set of possible labels. Reduced values of the subset accuracy, in particular, the subset accuracy of CAPEC predictions, thus, does not imply bad model behavior. Rather, they capture the reality of making simultaneous predictions of numerous correct labels in real world security conditions.

In general, the values of performance observed are consistent with the design objective of the system. These models are optimized to be as widely covered, interpretable and decision supporting as opposed to a limited optimization of a single metric. In this case, high recall and equal F1-scores will be desirable results because it means that the security analysts will have a full set of the attack and defense insights and will still be under full control of final decisions.

5.4 Results Analysis

The AI based penetration testing pipeline was tested with the help of vulnerability scan data of three different targets that is orca.org.bd, smartgrad.org and octobrain.org. These targets were chosen in a way that they could mirror a variety of production-facing web services with varying exposure and service configuration and vulnerability profile. In all the cases, the system was able to process Nmap scan results and to map the extracted Common Vulnerabilities and Exposures (CVEs) onto Common Weakness Enumerations (CWEs) and make multi-label predictions of CAPEC attack patterns, MITRE ATT&CK techniques and MITRE D3FEND defensive measures.

5.4.1 Results for orca.org.bd

The scan against orca.org.bd resulted in 20 distinct CVEs which are mostly related to the HTTP and HTTPS services that are available in ports 80 and 443. These weaknesses were picked out of the organized XML output created by Nmap through service version identification and the vulners script. The pipeline was able to process all the extracted CVEs, and CWE mappings were found in the curated master dataset.

Some of the vulnerabilities had similar CWE profiles causing homogeneous and repetitive predictive patterns. As an illustration, the vulnerabilities against the CWEs, including CWE-116 (Improper Encoding or Escaping of Output) and CWE-918 (Server-Side Request Forgery) yielded high confidence prediction across several taxonomies.

For CVE-2024-38474, the system predicted:

CAPEC: Variable manipulation (73), Brute Force (116), and XML External Entities (XXE) (250), which demonstrate several possible exploitation methods.

MITRE ATT&CK Techniques: T1027 (Obfuscated/Compressed Files or Information), T1562.003 (Impair Defenses) and T1574.006 (Hijack Execution Flow) which are post-exploitation and persistence-related techniques.

MITRE D3FEND: Harden, Isolate, and Evict with similar confidence scores, focus on a defense-in-depth approach instead of individual mitigation measure.

Equally, the CVE-2024-43204 that is related to SSRF-related weaknesses led to:

Embedding Scripts within Scripts (465) and XML Injection (219) are some of the predictions of CAPEC.
T1090.001 (Command and Scripting Interpreter: PowerShell) and T1528 (Data from Information Repositories), are also part of ATT&CK Techniques.
DEFEND recommendations focused on isolation, detection and execution analysis logging.

These findings show that the pipeline is useful to capture many-to-many relationships amongst weaknesses, attack paths and defensive controls, as are typical in real-world exploitation situations.

5.4.2 Results for smartgrad.org

The scan against smartgrad.org revealed 15 distinct CVEs in the services on ports 22 (SSH), 80 (HTTP) and 443 (HTTPS). The vulnerabilities that were extracted were a mixture of a web server vulnerability and the cryptographic or authentication vulnerability.

Based on 15 CVEs, 13 were identified to be mapable to CWEs with the master dataset. Two CVEs did not have any valid CWE mappings, and were transparently bypassed by the pipeline, demonstrating that it could deal with incomplete data on vulnerabilities without disruption of the analysis process.

There were a number of high-confidence predictions. As an example, CVE-2023-48795, which is associated with CWEs in the range of 345, 354, 693, 703 and 754, generated very powerful predictions in the CAPEC, such as CAPEC-145 (Check for Data Leakage) and CAPEC-463, which show data exposure and protocol abuse. Adversary behavior that was recommended by various ATT&CK techniques, including T1491, T1557.002, and T1584.002 implied adversary actions of infrastructure manipulation and man-in-the-middle. DEFEND outputs were biased towards Harden, Detect and protocol-oriented protection.

Such vulnerabilities in the web as CVE-2022-41741 and CVE-2022-41742 had the same results with both HTTP and HTTPS services, which reflects the fact that the system can generalize the similarity between the vulnerabilities found on the various service endpoints.

In general, the smartgrad.org case shows that the pipeline works well even in the situations when the set of vulnerabilities is smaller and contains incomplete mappings. The predictions that were generated were consistent, understandable and in accordance with the service exposure that was seen.

5.4.3 Results for octobrain.org

The exposure profile of the octobrain.org scan was the most diverse with 16 distinct CVEs being identified over a wider scope of open ports, namely 22, 80, 443, 3000, 8000, 9090, and 9100. This increased attack surface led to a more diverse and heterogeneous group of vulnerabilities including web services, application frameworks and administrative interfaces.

In contrast to the rest of the targets, one of the CVEs (CVE-2021-23017) was not found in the local data. The system automatically obtained its CWE mapping (CWE-193) in the API of NVD, registered the event, and archived the outcome in a separate unresolved-vulnerability log to be traced. This proves the soundness of the fallback mechanism and is guaranteed to be analytically complete and not require any manual work.

In the case of vulnerabilities that include CVE-2023-38408 and CVE-2023-48795, the system generated the same prediction patterns, which are in line with the patterns that existed in the case of the smartgrad.org, reinforcing the cross-target consistency. Nonetheless, more open services resulted in the forecasts of privilege escalation, configuration, and file permission. ATT&CK permissions of the local file, configuration repository and staged content were seen as common and defensive guidance was focused on isolation, detection, content quarantine and access modelling.

The results of the octobrain.org demonstrates the direct relationship between the greater diversity of services and the wider variety of anticipated attacker behavior and defensive responses, and still yielding structured and understandable results. Table 5.6 presents some examples of predicted output with CVE, CWE, CAPEC, ATT&CK Techniques, and D3FEND mappings.

CVE	CWE	CAPEC	TECHNIQUES	DEFEND
CVE-2023-38408	428, 664, 668	60 (0.79), 21 (0.78), 196 (0.78)	1539 (0.79), 1134.001 (0.79), 1550.004 (0.79)	Harden (0.80); Isolate (0.80); Evict (0.80)
CVE-2023-48795	345, 354, 693, 703, 754	145 (0.95), 463 (0.90), 701 (0.84)	1491 (0.83), 1557.002 (0.83), 1584.002 (0.83)	Harden (0.80); Detect (0.80); D3-SFCV (0.80)
CVE-2023-51385	707, 74, 77, 78	64 (0.90), 78 (0.90), 3 (0.89)	1027 (0.89), 1574.006 (0.88), 1562.003 (0.88)	Isolate (0.88); Detect (0.88); D3-DA (0.88)
CVE-2024-6387	362, 691	29 (0.78), 26 (0.77), 27 (0.53)	1505.005 (0.35), 1562.002 (0.35), 1562.008 (0.35)	Isolate (0.38); D3-LFP (0.38); Local File Permissions (0.38)
CVE-2025-26465	390, 703, 755	624 (0.46), 625 (0.46), 22 (0.41)	1083 (0.36), 1134 (0.33), 1134.001 (0.33)	Isolate (0.41); D3-LFP (0.41); Local File Permissions (0.41)

Table 5.6: Example of predicted output with CVE, CWE, CAPEC, ATT&CK Techniques, and D3FEND mappings

5.4.4 Cross-Case Observations

On all three targets, the pipeline was capable of deriving CVEs out of XML scan results without preprocessing, mapping CVEs to CWEs in a hybrid local/API model, and, finally, was able to produce ranked multi-label predictions across CAPEC, ATT&CK, and D3FEND and record cases that were not successfully matched in an unopaque manner.

The system was stable even when there are changes in the volume of vulnerabilities, the diversity of services, and the completeness of the datasets. Though the training data had high volume (around 150,000 records), the training time was close to one hour, inference in all the three case studies was light and could be deployed in practice.

More to the point, this part is concerned with prediction behavior and the results of analytical procedures instead of formal performance measures of models. In the model evaluation section, quantitative evaluation metrics, including that of precision, recall, and F1-score are presented independently of each other to keep a sharp distinction between the interpretation of results and model validation. Table 5.7 is the summary of the case study results.

Target Website	CVEs Detected	CWE Resolution	API Fallback Used	Dominant Services	Output Characteristics
orca.org.bd	20	All local	No	HTTP/HTTPS	Dense, consistent predictions
smartgrad.org	15	13 local, 2 skipped	No	SSH, HTTP, HTTPS	Smaller set, coherent outputs
octobrain.org	16	15 local, 1 API	Yes (CWE-193)	Multi-port services	Most diverse predictions

Table 5.7: Summary of Case Study Results

5.5 Final Design Adjustments

This section shows the last improvements carried out to the system design based on the performance evaluation outcomes presented in Section 5.1.2. These adjustments are not intended to change the basic design of the proposed AI-based penetration testing pipeline, but rather to improve its behavior, efficiency and usability according to strengths, weaknesses and practical issues that have been observed during the experimentation.

The initial system design, which was explained in Chapter 4, was a modular pipeline that converts the vulnerability scan output to higher-level security intelligence using CWE-based reasoning and multi-label machine learning models. The results of the evaluation ensured that the given core design is effective, stable, and can be generalized to various targets. Thus, the last design changes are aimed at the control of the final output, effectiveness of calculations, and clarity of work, and not at structural redesign.

Adjustment 1: Emphasis on CWE-Level Abstraction

The results of the performance evaluation reconfirmed that the operation at the CWE level has a great contribution to the presence of strong generalization and high predictive performance. The high recall and F1-score on CAPEC, MITRE ATT&CK, and MITRE D3FEND predictions point to the reduction of noise produced by CWE-specific differences and version-specific differences through CWE abstraction. Consequently, the ultimate design is explicitly committing CWE identifiers to be the sole and immutable feature representation employed to infer the models. Raw CVE text, service banners or exploit metadata are not introduced

into the learning process at all. This adaptation makes the process of feature engineering simpler, more interpretable and makes predictions consistent even in varied scanning conditions and software.

Adjustment 2: Controlled Multi-Label Output via Top-k Selection

In the initial testing stages, the models would sometimes generate a large count of projected labels on some kinds of vulnerabilities, especially those under the MITRE ATT&CK category. Although technically correct, such outputs decrease the interpretability and drown analysts. The last design is based on this observation via a top-k filtering mechanism probability based inference. Rather than showing all the labels that are predicted, the system is now able to display the top-k labels (e.g. top three) of those with the highest confidence score per taxonomy. This modification does not influence the training of the model or their evaluation measures but makes the output much easier to use and understand. According to this design choice, the purpose of the system as a decision-support tool is met, in which analysts have prioritized insights instead of comprehensive lists of labels.

Adjustment 3: Separation of Training and Inference Phases

As it was revealed during the evaluation process, model training is computationally intensive and needs a lot of time and resources in the event of a large dataset. Nonetheless, new scan results were found to be lightweight and quick to make inferences. In keeping with this action, the final design formally decouples the offline training phase and the online inference phase. Models are also trained after using the curated dataset and are then serialized to be repeated when it is used during live analyses. This modification also makes sure that this computational cost is incurred only in cases where retraining periodically is done, and not in the course of regular vulnerability checks. This separation renders the system implementable on typical analyst workstations and prevents recomputation which is not necessary.

Adjustment 4: Explicit Handling of Missing or Unresolvable Vulnerabilities

The analysis of various case studies showed that not all CVEs would be matched by CWE entries in the local database. Cases of incomplete predictions were not forcibly predicted, but the system was modified to simply record and do nothing with incomplete predictions. The last design incorporates a clear fallback process that tries external retrieval when conditions are right and records successful cases which cannot be resolved without producing speculative outputs. This change enhances the analytical integrity and makes the system consistent with the ethical practices of penetration testing by eschewing unsubstantiated assumptions.

Adjustment 5: Reinforcement of Interpretability Over Automation

The good predictive results accomplished in the evaluation may be an incentive to complete automation. Nonetheless, according to the design objectives of the system and ethical factors, the end design strengthens the human-in-the-loop functioning. The outputs are designed, prioritized, and put into context, and no automated exploitation or execution of attacks are carried out. This change will make sure that the system does not supersede the work of the analysts but instead assists in their reasoning to maintain accountability and minimize the chance of abuse.

Final Design Outcome

The end system design, having taken into account these modifications, is loyal to the original methodology as well as considering practical considerations that were uncovered in the course of evaluation. The refined design:

- Has good predictive abilities and generalization.
- Enhances the interpretability of output and usability by analysts.
- Minimizes the amount of computation used during deployment.
- Assures of transparency and ethical functioning.
- Very similar to real-life penetration testing processes.

All in all, the final design is a stable, efficient and practically deployable AI-assisted vulnerability analysis framework that was proven by empirical assessment and streamlined by continuous adjustment.

5.6 Statistical Analysis

This section will be a comparative statistical analysis of the proposed AI-driven penetration testing pipeline and the current AI-based penetration testing methods mentioned in the literature review. The analysis is done on the relative improvements by comparing and contrasting the results with prior research on manual penetration testing or real-world operational costs, instead of the improvements to be made in comparison to these. This makes the comparison remain consistent, evidence based, and grounded on published academic work.

5.6.1 Time Efficiency

A number of studies document severe time savings in penetration testing with the help of AI-based automation. According to Heim et al.(2023), (ChatGPT) seats penetration testing saved up to 30 percent of manual effort, with reconnaissance and

configuration checks being the most automated (at 17 and 25 percent, respectively) [8]. On the same note, More and Rohela (2023)[12] have found that AI-supported automation has significantly increased vulnerability assessment workflows by 45% effectively. The time savings of about 45 per cent in Railkar and Joshi (2023)[13], and a 30 per cent reduction in the time of analysis in Hu et al. (2023)[9] were reported by these two research works with deep reinforcement learning employed in attack path discovery.

Comparatively, the proposed system can extract the vulnerabilities, map the CWE, and predict the attack-defense in several minutes after the scan data is provided. This is an offline process as it requires intense calculations to train the model. In the process of inference, generation of predictions is real-time. This puts the suggested system at the high-end of time efficiency statistically among AI-assisted penetration testing systems based on supervised or hybrid learning and not based on the sequential execution of attacks.

5.6.2 Automation and Human Effort Reduction

The current AI-based systems often focus on attack implementation or path planning automation. Other reinforcement learning-based methods like IAPTS[14] and Shennina[10] achieve improvement in automation between 40% and 50 percent, but they must be controlled by humans in case of ethical control, validation, and multi-step reasoning.

The difference with the proposed system is that it does not aim at automating exploitation. Rather, it automates the analysis of the argument, converting vulnerabilities into organised attack as well as defense intelligence. This minimizes the interpretation and mapping work of the analysts as opposed to implementation. The proposed method eliminates generative uncertainty (using structured taxonomies and supervised learning), unlike the systems based on LLM that have a maximum hallucination rate of 18% [7].

5.6.3 Computational Cost and Scalability

Scalability and computational overhead are also among the challenges found in several studies. According to McKinnel et al. (2023)[11], both POMDP-based and attack-graph-based systems have exponential growth of execution time when the size of the network contains more than 50 hosts. Attack optimization methods based on the Genetic Algorithms could have a 500 percent increase, although execution times remained high. The more advanced methods of deep reinforcement, e.g., Hu et al. (2023)[9] and Samad et al. (2023)[14], also experienced higher costs associated with computations since the simulation and policy learning were repeated.

Conversely, the suggested system does not combine training and inference. Large-scale data training is expensive, but linear inferences are inexpensive as the number of vulnerabilities increases. This design will not involve repetitive simulation, expansion of attack graphs, or policy iteration, which makes it more appropriate in cases of repeated evaluation of multiple targets.

5.6.4 Accuracy and Reliability

Accuracy values in previous studies are reported to be very diverse. Systems relying on reinforcement learning generally achieve 85–90% success in attack path selection, whereas those relying on deep learning intrusion detection systems achieve more than 95 percent accuracy in controlled datasets. Nevertheless, several researches point at false positives, hallucinations, or complex attack chain failures.

The generated system has high micro-averaged F1 scores in prediction on CAPEC, MITRE ATT&CK, and MITRE D3FEND. Although these values are high, they represent prediction of established structured relationships as opposed to dynamic exploitation. The proposed system works out to offer more predictable and explainable outputs as compared with the LLM-based systems that have an error rate ranging between 12-18% because it was developed based on taxonomy. Table 5.8 demonstrates the statistical comparison of the proposed system and literature AI-based penetration testing methods.

Study / Approach	**Time Reduction**	**Automation Focus**	**Computational Cost**	**Reported Accuracy / Success**
Heim et al. (2023)[8]	∼30%	Reconnaissance & analysis	Low–Medium	∼75% agreement with experts
More & Rohela (2023)[12]	∼45%	VAPT automation	Medium	∼95% vulnerability detection
Hu et al. (2023)[9]	∼30%	Attack path optimization	High	∼86% path accuracy
Samad et al. (2023)[14]	∼40–50%	RL-based attack planning	High	∼90% optimal path accuracy
Karagiannis et al. (2023)[10]	∼40%	Automated cyberattack simulation	High	∼85% vulnerability discovery
Proposed System	Near-instant inference	Analytical reasoning (CWE→Attack→Defense)	High (training), Low (inference)	High F1-scores on structured taxonomies

Table 5.8: Statistical comparison of the proposed system and literature AI-based penetration testing methods

5.7 Comparisons and Relationships

The section compares the suggested AI-based penetration testing framework with the literature reviewed in Chapter 2, based on the scope of its operations, the cost of computation, the efficiency in terms of time, the degree of automation, and the purpose of analysis. The comparison does not mention the superiority but it indicates the position occupied by the proposed approach in the wider research on penetration testing.

The majority of current AI-based Penetration Testing systems focus on attack executions, path-planning, or autonomous exploitation and most of those are based on

Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) or Markov models. Artificial intelligence can be used to study the behavior of attackers including Heim et al. (2023)[8], Happe and Cito (2023)[7], Hu et al. (2023)[9], and Ghanem and Chen (2019)[2]. Although these methods prove to be very efficient in a controlled setting, they are often high-computation, low-scalability, unethical and lack interpretability particularly when used with real-world production infrastructures.

The system proposed, on the contrary, is a planned decision-support and analytical system, and not an independent attack engine. It aims at mapping observable vulnerabilities (CVEs → CWEs) into standardized attack and defense knowledge bases (CAPEC, MITRE ATT&CK, and MITRE D3FEND) instead of learning sequential attack strategies. The design decision makes the system more consistent with risk interpretation and threat modeling and defensive planning, and less consistent with exploitation automation.

One of the most critical relationships is formed when the computational features are compared. The converging systems of RL-based reported in McKinnel et al. (2023)[11], Hu et al. (2023)[9], and Samad et al. (2023)[14] have massive simulation set-ups, repeated training cycles and high-performance computing platforms to converge. Contrary to that, the supervised multi-label learning method proposed in this model transfers the computational cost to offline training. Even though the process of training took nearly one hour on one of the consumer-grade laptops, the process of inference during live analysis was lightweight and took seconds. Such a division of training cost and inference efficiency of a system renders it more appropriate to recurrent operational use without the specialized hardware.

Regarding data, most of the previous literature clearly states that the absence of a standard set of data is a significant drawback. McKinnell, et al. (2023)[11], Railkar and Joshi (2023)[13] and Raza (2024)[17] focus on fragmentation on both datasets and environments. The suggested system has a direct answer to this problem since it is based on standardized and widely accepted taxonomies (CWE, CAPEC, ATT&CK, D3FEND), allowing the use of the same reasoning regardless of the target without the need to retrain the system to the specifics of the environment.

The other significant difference is the ethical limits and practicality in the real world. Some of the LLM-based and RL-based methods in the articles of Heim et al. (2023)[8] and Happe and Cito (2023)[7] face hallucinations, unsafe execution of commands, or ethical limitations on testing in the real world. These risks are prevented by the proposed system through not making attacks, not creating exploit payloads, and not circumventing protections. This can be well demonstrated in the readloudly.com case study: the system did not yield any predictions when it could not enumerate vulnerabilities at the infrastructure level, but it should have done so instead of making speculative ones.

On the whole, the suggested framework can be used to supplement, but not to substitute the current AI-based penetration testing schemes. It bridges the gap between the raw vulnerability scanning tools and entirely autonomous attack system by providing a more structured, explainable and scalable approach to converting the evidence of vulnerabilities into actionable security intelligence. Table 5.9 is comparative summary between existing works and our works.

Aspect	**Proposed System**	**RL / DRL-Based Systems (Hu et al. (2023)[9], Samad et al. (2023)[14], Ghanem & Chen (2019)[2])**	**LLM-Based Systems (Heim et al. (2023)[8], Happe & Cito (2023)[7])**
Primary Goal	Vulnerability interpretation & decision support	Automated attack planning & execution	AI-assisted exploitation & reasoning
Learning Paradigm	Supervised multi-label learning	Reinforcement / Deep Reinforcement Learning	Large Language Models
Input Basis	CVE → CWE mappings	Network state & attack graphs	Natural language + system feedback
Output	CAPEC, ATT&CK, D3FEND predictions	Attack paths, exploits, rewards	Commands, scripts, attack steps
Execution of Attacks	No	Yes	Yes (partially)
Ethical Risk	Low	High	Medium–High
Training Cost	High (offline only)	Very high (continuous)	Low–Medium
Inference Cost	Low	Medium–High	Medium
Hardware Requirement	Consumer-grade laptop	Often requires high-performance systems	Moderate
Scalability	High	Limited by environment size	Limited by prompt quality
Real-World Deployment	Strong	Constrained	Constrained

Table 5.9: Comparative Summary Table

5.8 Simulation-Based Validation

To further substantiate the practical usefulness of the suggested AI-based penetration testing pipeline, controlled simulations of some of the forecasted vulnerabilities were carried. These simulations were not aimed at exploiting real systems but seeing whether the predicted attack patterns and adversary techniques match with the expected behaviors when similar weaknesses are replicated to a controlled environment. This step will offer another step of validation over a statistical model analysis since it connects predictions to the technical behaviors that may be observed.

The malware simulation tests were conducted in a highly susceptible, locally installed web-based test infrastructure, which was created to simulate typical input-

processing,validation and processing vulnerabilities. The simulated scenarios were formed to correspond to the fundamental weakness mentioned by the relevant CWE, and not to duplicate the entire vulnerability on the real world. This strategy is ethical and at the same time provides a direct view of how individual types of weaknesses occur during the input processing, request processing, and the execution of the back-end logic.

In CVE-2024-38474 the simulation centered on unsafe output rendering which is represented as CWE-116 (Improper Encoding or Escaping of Output). The information inputted by users was reflected back to the response without encoding. This model forecasted CAPEC-73 (Variable Manipulation) and an ATT&CK scripting-based technique. The input that was injected during simulation was not encoded in the output, which proves that the predicted attack pattern is an accurate representation of the observed weakness behavior. This confirms the capability of the model to correlate the weaknesses in the encoding of output with their attacks that are based on manipulation.

In CVE-2024-43204 (related to CWE-918 (Server-Side Request Forgery)), the simulation involved the study of the effect of input provided by the user to outbound requests. The MITRE ATT&CK model suggested predicts an injection style pattern of CAPEC and T1090.001. In the process of execution, a designed input triggered the application logic to make an external resource request, which confirms that the forecasted behavior is true to the actual manifestation of SSRF-style weaknesses.

In the case of CVE-2023-38709 that is associated with the CWE-20 (Improper Input Validation) the simulated input logic was inconsistent. There was no strict rule of validation and the app accepted some broken inputs and rejected others. The anticipated CAPEC and ATT&CK methods focused on deception and inappropriate handling. The given behavior proved that the lack of consistency in validation results in an exploitable ambiguity, which proved the accuracy of the anticipated attack pattern.

CVE-2024-7008 simulation, which is related to CWE-79 (Cross-Site Scripting), was devoted to unsafe rendering of HTML-type input. The model forecasted CAPEC-63 (XSS) and a scripting-associated ATT &CK method. In simulation, the HTML like content injected was displayed verbatim in the page output indicating unsafe reflection and validating the consistency in attack pattern prediction and observed execution pattern.

In case of CVE-2024-52969 that is CWE-89 (SQL Injection), the simulation was analyzed with respect to user input and query construction. The input was concatenated in a query-like string, which was not sanitized by the application logic. The model forecasted CAPEC-66 (SQL Injection) and a technique of exploiting the server. This was a source of the observed behavior and this prediction was validated by the fact that the model is accurate in identifying risks of injection according to the characteristics of the weaknesses.

The simulation of CVE-2024-2602 dealt with CWE-22 (Path Traversal). Without being normalized, user input was added to a base file path. The model forecasted CAPEC-126 (Path Traversal) and a data access method. In simulation, the desired path structure was distorted by designed input, which showed naive path construc-

tion and supported the predicted pattern of attack.

Lastly, the CVE-2024-48889 (related to CWE-78 Command Injection), was modeled by composing a command string by simply concatenating user input. CAPEC-248 (Command Injection) and command execution ATT&CK technique were predicted using the model. The model was inferred to be correct since the simulation proved that unsafe command construction logic might be affected by user input.

In the controlled environment, the behaviors observed were predicted by the CAPEC attack patterns and the MITRE ATT&CK techniques across all the simulated cases. These simulations indicate that the predictions of the model are not abstract associations, but realistic effects on the execution level in the case of similar weaknesses. Notably, the simulations were carried out in a closed and non exploitative manner of the live systems and thus ethical concerns were addressed during the validation process.

This validation of the system through a simulation also supplements the quantitative assessment done in the previous section since it shows that system predictions are consistent with realistic vulnerability behavior. In combination, the statistically evaluated performance measures and the simulated model reinforce the perspective of the capability of the system to facilitate the penetration testing analysis and the process of defensive decision-making in the real world. Table 5.10 is the summary of simulation based validation.

CVE ID	**CWE**	**Predicted CAPEC (simple)**	**Predicted ATT&CK (simple)**	**Observed Behavior in a controlled web-based simulation environment**	**Validation**
CVE-2024-38474	CWE-116	CAPEC-73 (Variable Manipulation)	T1059	Input reflected without encoding	✓ Valid
CVE-2024-43204	CWE-918	CAPEC-465 (Request/Injection-style)	T1090.001	User input triggers external resource request	✓ Valid
CVE-2023-38709	CWE-20	CAPEC-182 (Injection/Improper Handling)	T1036.001	Input handling inconsistent (accepted/rejected)	✓ Valid
CVE-2024-7008	CWE-79	CAPEC-63 (XSS)	T1059.007	HTML-like input renders as page content (unsafe reflection)	✓ Valid
CVE-2024-52969	CWE-89	CAPEC-66 (SQL Injection)	T1190	Query string constructed directly from input (unsafe pattern)	✓ Valid
CVE-2024-2602	CWE-22	CAPEC-126 (Path Traversal)	T1005	User input appended to base path (naive path build)	✓ Valid
CVE-2024-48889	CWE-78	CAPEC-248 (Command Injection)	T1059	Command string built by concatenating user input	✓ Valid

Table 5.10: Simulation-Based Validation Summary

5.9 Discussion and Limitations

This part is based on the phenomenological considerations made on the basis of the experimental testing of the suggested AI-driven penetration testing pipeline. Instead of using the metrics of numerical performance, the discussion focuses on the behaviour of the system under real-world conditions, what the system can consistently do, and where its operation is limited by external considerations like infrastructure-level defenses.

In the three main products of interest in the case study (orca.org.bd, smartgrad.org, and octobrain.org), the system showed predictability and consistency in behavior when the vulnerability information regarding its service level was exposed to the outside. Overall, CVEs were found in the output of Nmap scans in all three instances and were successfully resolved to give relevant CWE identifiers and meaningful multi-label predictions of CAPEC attack patterns, MITRE ATT&CK techniques, and MITRE D3FEND defensive measures. These findings verify that the design choice to reason at the CWE level and not the CVE level directly makes it possible to generalize effectively across the various hosts, services, and software versions. Weaknesses with similar weakness attributes always generated similar attack and defense projections, even in the case that they are observed in other systems.

The other notable insight is that the system is purposely aimed at the coverage rather than precision that is very narrow. The evaluation results demonstrate this behavior as micro-averaged recall and F1-scores are high in all prediction categories. This practically implies that the system is constructed in such a way as to not miss plausible attack paths or defensive action, even at the expense of considerations of wider prediction sets. This would be preferred in the situations of supporting penetration testing and defensive planning, since in such way, analysts would be able to filter, rank, and contextualise predictions instead of risking to miss important security risks.

Besides the three main case studies an observational scan was performed against readloudly.com to observe the behaviour of the pipeline under the current infrastructure level protections. In comparison to its main targets, readloudly.com is under the protection of Cloudflare, which is a reverse proxy and a content delivery network. Because of this, Nmap was capable of only determining Cloudflare proxy services on standard web ports, and not the underlying web server software, application frameworks, or version data. As a result, the vulners script could not correlate the detected services with the known CVEs, and no vulnerability identifiers were generated.

This fact is not the failure of the suggested system. Rather, it puts out a notable and significant boundary condition. The pipeline is based on observed evidence of vulnerabilities that has been created during scanning. In the case of upstream tools that are deliberately blocked in revealing the service details, the system will make the right decision of avoiding the generation of predictions instead of giving speculative or confusing outputs. This action supports both ethical and analytical construction of the structure, which does not seek to avoid security measures and the implied vulnerabilities in the absence of supporting facts.

The primary case studies can also be compared with the readloudly.com observation that can be used as the helpful contrast. Whereas orca.org.bd, smartgrad.org, and octobrain.org have revealed enough information to enumerate and analyze the vulnerabilities, readloudy.com shows that modern defensive infrastructure can dramatically lower externally visible attack surfaces. This difference points to the importance of defense-in-depth strategies, where the mitigation of risks is conducted not alone by patching the vulnerabilities but also by architectural-based controls, which restrict the reconnaissance opportunities.

On the computational side, the experiments are valid that the most resource-intensive stage of the system is model training. The training on the entire dataset took almost an hour using normal workstation hardware. This is however only incurred in the offline training. After training, the models did inference well and the generation of prediction of all the targets evaluated could be done without any perceivable delay. This distinction between training cost and inference efficiency is in favor of the applicability of the system to repeated operation.

In general, the empirical findings of the experimental analysis indicate that the suggested system is predictable, transparent, and responsible in both conducive and restricted conditions. It generates expressive and understandable outputs where vulnerability evidence is present, it does not conjecture in the absence of evidence, and it works in harmony with the current security measures. The purpose of this is to be an AI-assisted analytical framework to support penetration testing and defensive decision-making, and not an automated exploitation system which is reflected in these features.

Chapter 6

Conclusion

6.1 Future Work

Although the suggested AI-based penetration testing support system has a high level of analytical ability, as well as realistic applicability, some extensions can be considered to expand its range, authenticity, and usefulness. One future work direction would be the combination of sandbox-based simulation of exploitation and attack. The existing system is concerned with predicting attack patterns and defense mechanisms analytically in accordance with the observed vulnerabilities. The extensions that can be made in future involve controlled sandboxing where the anticipated attack patterns are run in a secure and isolated environment. This form of simulation would make it possible to validate the CAPEC and MITRE ATT&CK techniques being predicted by means of observable system behavior which would enhance the connection between predictive intelligence and practical exploitation dynamics without transgressing ethical boundaries. The other extension of significant importance is authenticated and internal scanning situations. The current system works using externally visible vulnerability data that has been acquired using an unauthenticated network scan. In environments with actual organizations, internal testing and authenticated tests usually can expose more configuration errors, privilege increasing routes, and logic-level weaknesses. It would be possible to reason over a more detailed attack surface to make more detailed attack and defense predictions and to extend the pipeline to consume authenticated scan results, internal asset inventories, or configuration-level data. Another future direction of work is evolution of datasets and adaptability in real time. The models used in this work are trained based on a curated dataset of large size, based on publicly available vulnerability and attack taxonomies. Nevertheless, cybersecurity knowledge is dynamic and new CVEs, CWEs, attack and defense methods are implemented. Future efforts may be on the incremental or periodical retraining schemes that may take in new published vulnerability information, and real-time threat intelligence feeds. This would assist in keeping the predictions up to date and in line with the current changes in the attacker behavior and defensive practices.

6.2 Conclusion

This thesis introduced an AI-assisted vulnerability analysis and penetration testing support framework used to fill the gap between the raw vulnerability scan output and actual security intelligence. The proposed system was able to map the detected vulnerabilities to corresponding CAPEC attack patterns, MITRE ATT&CK techniques, and MITRE D3FEND defensive measures, using supervised multi-label machine learning by operating at the Common Weakness Enumeration (CWE) level. As a result of this experimental assessment on a series of real-world and controlled case studies, it was found that the system is capable of reliably processing network scan findings, remediating vulnerabilities through standardized taxonomies, and producing interpretable and ranked predictions that can be used to support decisions. The findings supported the view that the multi-label models using the Random Forest architecture had high predictive performance, especially in recall and F1-score, which is consistent with the goal of the system to cover as many as possible instead of to focus on a small number of security analysis cases. The study also made significant practical observations besides predictive performance. This system is predictable when the evidence of vulnerability is observable externally, and infrastructure-level restrictions like reverse proxies and CDN can be consciously used to restrict the effects of enumeration. This helps to establish the limits of the ethical design of the system and explain its scope of operation as an analysis support system and not an engine of exploitation. All in all, the suggested framework shows that AI-based analysis supported by standardized cybersecurity body of knowledge and guided by the aspect of interpretability can make a substantial change to the vulnerability assessment process. The work forms a scalable, transparent, and ethically consistent support of penetration testing based on AI-assistance, which can form a strong basis in the field of future extensions that would involve controlled simulated attack, authenticated scanning, and the integration of adaptive threat intelligence.

Bibliography

[1] Y. Stefinko, A. Piskozub, and R. Banakh, "Manual and automated penetration testing: Benefits and drawbacks," *Modern Tendency*, pp. 488–491, Feb. 2016. DOI: 10.1109/TCSET.2016.7452095.

[2] M. C. Ghanem and T. M. Chen, "Reinforcement learning for efficient network penetration testing," *School of Mathematics Computer Science and Engineering, University of London*, vol. 1, no. 1, pp. 1–10, 2019. DOI: 10.1007/xyz1234.

[3] S. Tjoa and P. Kieseberg, "Penetration testing artificial intelligence," *Cybersecurity Journal*, vol. 1, pp. 1–15, 2020.

[4] D. Antonelli, R. Cascella, G. Perrone, S. P. Romano, and A. Schiano, "Leveraging ai to optimize website structure discovery during penetration testing," *arXiv preprint*, 2021. arXiv: 2101.07223v1.

[5] P. Garrad and S. Unnikrishnan, "Artificial intelligence in penetration testing of a connected and autonomous vehicle network," in *Conference Paper*, 2022.

[6] K. Z. Azar et al., "Fuzz, penetration, and ai testing for soc security verification: Challenges and solutions," in *Conference Proceedings*, 2023.

[7] A. Happe and J. Cito, "Getting pwn'd by ai: Penetration testing with large language models," in *Conference Proceedings*, TU Wien, 2023.

[8] M. P. Heim, N. Starckjohann, and M. Torgersen, "The convergence of ai and cybersecurity: An examination of chatgpt's role in penetration testing and its ethical and legal implications," Bachelor's thesis, Norwegian University of Science and Technology, May 2023.

[9] Z. Hu, R. Beuran, and Y. Tan, "Automated penetration testing using deep reinforcement learning," *Japan Advanced Institute of Science and Technology*, 2023.

[10] S. Karagiannis et al., "Ai-powered penetration testing using shennina: From simulation to validation," in *Conference Proceedings*, Ionian University, University of Naples Federico II, Montimage, PDM Lisbon, 2023.

[11] D. R. McKinnel, T. Dargahi, A. Dehghantanha, and K.-K. R. Choo, "A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment," *University of Salford, University of Guelph, University of Texas at San Antonio*, 2023.

[12] S. More and A. Rohela, "Vulnerability assessment and penetration testing through artificial intelligence," *Department of Information Technology, SJCEM*, 2023.

[13] D. Railkar and S. Joshi, "A comprehensive literature review of artificial intelligent practices in the field of penetration testing," in *Advances in Cybersecurity*, Springer, 2023. DOI: 10.1007/978-981-19-6581-4_7.

[14] A. Samad, S. Altaf, and M. J. Arshad, "Advancements in automated penetration testing for iot security by leveraging reinforcement learning," *University of Engineering & Technology Lahore*, 2023.

[15] U. Governance I., *Data breaches and cyber attacks – usa report 2024*, https://www.itgovernanceusa.com/blog/data-breaches-and-cyber-attacks-in-2024-in-the-usa, Accessed: 2024-06-19, Jun. 2024.

[16] S. R. Gudimetla, "Enhancing penetration testing with machine learning and artificial intelligence: A comprehensive analysis," *International Journal of Innovative Research in Science, Engineering and Technology*, 2024. DOI: 10.15680/IJIRSET.2024.1307003.

[17] H. Raza, "Systematic literature review of challenges and ai contributions in penetration testing," Master's thesis, Luleå University of Technology, 2024.

[18] S. Alder, *Healthcare data breach statistics*, https://www.hipaajournal.com/healthcare-data-breach-statistics/, Accessed: 2025-10-01, Oct. 2025.

[19] E. Bonnie, *110+ of the latest data breach statistics to know for 2026 & beyond*, https://secureframe.com/blog/data-breach-statistics, Accessed: 2025-09-24, Sep. 2025.

[20] Fortinet, *Top cybersecurity statistics: Facts, stats and breaches for 2025*, https://www.fortinet.com/resources/cyberglossary/cybersecurity-statistics, 2025.

[21] J. Nadeau, *83% of organizations reported insider attacks in 2024*, https://www.ibm.com/think/insights/83-percent-organizations-reported-insider-threats-2024, Accessed: 2025-10-01, Oct. 2025.

[22] SentinelOne, *Key cyber security statistics for 2025*, https://www.sentinelone.com/cybersecurity-101/cybersecurity/cyber-security-statistics/, Accessed: 2025-10-07, Oct. 2025.

[23] P. Spencer and P. Spencer, *2024 cybersecurity and compliance landscape: 50 critical statistics shaping our digital future*, https://www.kiteworks.com/cybersecurity-risk-management/2024-cybersecurity-landscape-50-critical-statistics/, Accessed: 2025-09-24, Sep. 2025.

[24] J. M.-. C. Strategist, *195 cybersecurity statistics (updated june-2025)*, https://www.brightdefense.com/resources/cybersecurity-statistics/, Accessed: 2025-06-29, Jun. 2025.