A Systematic Literature Review and Meta-analysis on Artificial Intelligence in Penetration Testing and Vulnerability Assessment

Dean Richard McKinnel[1], Tooska Dargahi[1], Ali Dehghantanha[2], Kim-Kwang Raymond Choo[3]

1- Department of Computer Science, University of Salford, Manchester, UK
2- Security of Advanced Systems Lab, School of Computer Science, University of Guelph, Ontario, Canada
3- Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249, USA

d.r.mckinnel@edu.salford.ac.uk, T.Dargahi@Salford.ac.uk, A.Dehghan@UoGuelph.ca, raymond.choo@fulbrightmail.org

Abstract

Vulnerability assessment (e.g., vulnerability identification and exploitation; also referred to as penetration testing) is a relatively mature industry, although attempting to keep pace with the diversity of computing and digital devices that need to be examined is challenging. Hence, there has been interest in exploring the potential of artificial intelligence in enhancing penetration testing and vulnerability identification of systems, as evidenced by the systematic literature review performed in this paper. In this review, we focus only on empirical papers, and perform a meta-analysis of the located literature. Based on the findings, we identify a number of potential research challenges and opportunities, such as scalability and the need for real-time identification of exploitable vulnerabilities.

Keywords: Penetration Testing; Vulnerability Assessment; Artificial Intelligence; Systematic Literature Review, Machine Learning, Meta-Analysis

1. Introduction

Artificial intelligence (AI) is a fairly established research area. For example, AI was reportedly first formally established as a scientific research field by Dartmouth college's summer research project in 1956, which attempted to enable computational devices to solve complex problems [1]. In recent years, there have been increasing interests and attempts to utilize and adapt AI, such as machine learning (ML) techniques, in various disciplines, such as engineering, science and business, and everyday applications [2],[3]. Ayodele [4], for example, attempts to categorize existing ML algorithms, based on their outcomes.

Similar to other areas of research, there are a number of potential research challenges and opportunities relating to modern AI and ML techniques [5], [6], [7], including in cyber security applications [8], [9], [10]. In recent times, there have also been attempts to integrate or utilize AI and ML techniques (in this paper, both AI and ML terminologies are used interchangeably) in identifying vulnerabilities in systems that can be exploited, for example to facilitate covert data exfiltration.

Vulnerability identification and exploitation, also referred to as vulnerability assessment or penetration testing (pentesting) in the literature, comprises a range of different activities that can also be used to inform or enhance the mitigation strategies for a system [11],[12]. For

example, pentesting (in this paper, pentesting, vulnerability assessment, and vulnerability identification and exploitation, are also used interchangeably) can be used to facilitate information gathering (reconnaissance) to understand and assess the current state of a system, or more maliciously to actively exploit a system or network and gain unauthorised / active persistence access, for example using backdoors, to the target / vulnerable systems [13]. It is increasingly challenging to perform penetration tests due to the complexity and heightened security of the systems, as well as increased security awareness of system, software and hardware developers and security professionals [14]. Hence, one observed trend is in intelligence-oriented pentesting [15].

While there have been attempts to leverage AI and ML techniques in pentesting activities [18], [19], there is a lack of a systematic literature review (SLRs) or meta-analysis of existing literature. For example, Dogan et al. [16] and Hydara [17] performed SLRs on various aspects of penetration testing, but not AI/ML. Hence, this is a gap we seek to contribute to in this paper. Specifically, in this paper, we will systematically analyse existing AI / ML techniques utilised for penetration testing, focusing on the different applications and their performance. Moreover, we perform a meta-analysis of the located materials, and group these different algorithms, techniques and frameworks. Based on the findings, we conclude the paper with potential research opportunities.

In the next section, we will describe our literature review methodology. In Sections 3 and 4, we discuss the findings from our literature review and potential research agenda, before concluding the paper in Section 5.

2. Research Methodology

To aid in the collection of relevant studies, a set of search strings was constructed by extrapolating the research key terms, such as "penetration testing" and "pentesting". Different combinations of the title, abstract and keywords were manually assessed with the aim of focusing only on papers most relevant to the study, acquired from research databases listed in Table 1. For instance, when we searched using Google Scholar, we included "-*site:books.google.com*" to remove any books from the search. We also excluded citations and patents during our Google Scholar searches (see also Table 1). The search queries were conducted on 16th of February 2018. The located papers then underwent a snowballing process, in which references of these located papers were studied to locate find other relevant papers. The snowballing process occurred for both backward and forward lookups, and was finalised once we determined that all, if not most, papers relating to this study were found. Following this initial dataset construction, inclusion and exclusion criteria were applied to the entire dataset that allowed the dataset to be refined to only those most relevant papers (see also Table 2).

Table 1: Search query variation within each database

Database Name	Search Query	Return Value
Google Scholar	*"penetration testing" OR pentesting "penetration-testing" OR "vulnerability assessment" AND "artificial intelligence" OR "artificial-intelligence" OR "neural network" OR "neural-network" OR AI -"geotechnical" -site:books.google.com*	163
IEEE Explore (Title only as metadata gave ambiguous results)	*(("Publication Title":"penetration testing" OR "pentesting" OR "penetration-testing" OR "vulnerability assessment") AND ("Publication Title":"artificial intelligence" OR "machine learning" OR "machine-learning" OR "neural network" OR "neural-network" OR "artificial-intelligence" OR "AI"))*	2

ACM Digital Library	*(+"penetration-testing" +OR +pentesting +OR +"penetration testing" +OR +"vulnerability assessment" +AND +"artificial intelligence" +OR +AI +OR +"neural-networks" +OR +"neural networks" +OR +"machine learning" +OR +"machine-learning")*	11
Science Direct	*("machine learning" OR "machine-learning" OR ai OR "neural network" OR "neural-network" OR "artificial intelligence" OR "artificial-intelligence") AND ("penetration testing" OR "penetration-testing" OR pentesting OR "vulnerability assessment" OR "vulnerability-assessment") [All Sources(Computer Science)].*	278
Web of Science	*TS=("machine learning" OR "machine-learning" OR ai OR "neural network" OR "neural-network" OR "artificial intelligence" OR "artificial-intelligence") AND TS=("penetration testing" OR "penetration-testing" OR pentesting OR "vulnerability assessment" OR "vulnerability-assessment")*	58
Scopus	*TITLE-ABS-KEY("machine learning" OR "machine-learning" OR AI OR "neural network" OR "neural-network" OR "artificial intelligence" OR "artificial-intelligence") AND TITLE-ABS-KEY("penetration testing" OR "penetration-testing" OR pentesting OR "vulnerability assessment" OR "vulnerability-assessment")*	151

2.1 Filtration of Relevant Studies

Duplicate papers found within each of the database datasets were removed. To ensure the relevance of the papers, both inclusion and exclusion criteria were established. Specifically,

- The paper must focus on AI / ML, with direct applications to penetration testing or vulnerability assessment.
- The paper must include an empirical study, where data is collected and analysed, for example in case studies or technical evaluations of current AI / ML techniques for penetration testing and vulnerability assessment. In other words, there must be some measurable / prediction outcomes that can be quantified and compared with other outcomes from other approaches or techniques. Examples of suitable prediction outcomes include false positive rate, and learning time required for the system.
- The paper will also be included if it contains some system that incorporates partial applications of AI / ML.
- The paper must be either a peered review conference or journal paper, and published in the English language.

Studies that contain AI but do not adequately encompass penetration testing or vulnerability assessment (e.g., making generalisations of the AI / ML model having applications towards penetration testing and vulnerability assessment) were excluded. For example, algorithms that do not inherently show aspects of “learning” were excluded from our study – see also Table 2.

Table 2: Inclusion and exclusion criteria

Inclusion Criteria
1. The paper must focus on AI / ML, with direct applications to penetration testing or vulnerability assessment.
2. The paper must include an empirical study, where data is collected and analysed, for example in case studies or technical evaluations of current AI / ML techniques for penetration testing and vulnerability assessment. In other words, there must be some measurable / prediction outcomes that can be quantified and compared with other outcomes from other approaches or techniques. Examples of suitable prediction outcomes include false positive rate, and learning time required for the system.

3. The paper will also be included if it contains some system that incorporates partial applications of AI / ML. The paper must be either a peered review conference or journal paper, and published in the English language.
Exclusion Criteria
1. Studies that focus on areas other than AI even though they may related to penetration testing and/or vulnerability assessment, such as automated code generation or linear code-based algorithms that do not incorporate any AI techniques. 2. Unpublished papers that are uploaded to archive or the conference version of an extended journal paper. 3. Papers in a different language from that of English and grey literature that is not recognised as a reputable source of research (e.g., predatory conferences and journals).

2.2. Quality Assessment

To ensure that all the primary studies found contained suitable information for analysis and were relevant to the research area, a quality assessment was constructed. This quality assessment was built on the guidelines of Kitchenham and Charters [20], recommended by Hosseini et al. [21]. These guidelines were altered to match the context of this research area, but for the most part, the structure of the questions is still similar. The quality assessment is broken down into smaller stages to systematically check the quality of each of the papers in turn. Every paper in this study was reviewed using the following criteria to determine if the quality of the papers met that needed for analysis. Only once a paper had met all the criterion below would it be included for analysis.

Stage 1: Algorithm construction or application – In order to be applicable to the study, the paper must include the process of building or applying an AI / ML concept to an area of penetration testing or vulnerability assessment. Furthermore, if the AL / ML algorithm or concept undergoes a training period, the training period must be sustainable and include a varied range of training data so that it can be adequately compared. Similar to approaches based on neural networks, ML, genetic algorithms and other pattern identifying approaches were taxonomized under the broad umbrella of AI in this study. The taxonomy is justified by referring purposefully to the Oxford English Dictionary definition that AI "*is the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision making, and translation between languages*" [22]. This definition outlines the general abstract classification of AI in the wider community, therefore in this context the classification of other concepts like that of ML and neural networks for example, should be deemed acceptable.

Stage 2: Context – Context data must be supplied within the paper, such as the granular details of the programming language to execute the paper's proposed solution used to process the data within the AI concept.

Stage 3: Algorithm details – This criterion is the most quintessential for analysis as it involves the analysis of both independent and dependent variables and whether they are clearly established and reported in the paper itself. If a paper is evaluated quantitatively, the paper must contain both and independent and dependent variable to be deemed applicable.

Stage 4: Data – Information regarding the datasets used for the assessment of the model must be explicitly outlined. If private datasets are used, it is necessary to explain the composition of this data and how it has been standardised for use in an AI scenario.

Stage 5: Performance – The performance of each of the models, algorithms or applications must be measured and presented accurately within the paper.

2.3. Data Extraction

Studies that had proven to fare well in the quality assessment criteria were then passed for data extraction. To systemically carry out the data extraction a data extraction form was created comprising three sections, namely: contextual data, qualitative data and quantitative data. If the data extraction form was adequately populated with relevant and comparable qualitative and quantitative data, it would be then deemed suitable for further analysis.

Context data: This data includes data of context such as the type of penetration testing or vulnerability assessment domain that the study is focused on, as well as the aims of the study and various other types of data pertaining to the context.

Qualitative data: The qualitative data section regards the overall findings and conclusions of the study itself. Due to the nature of some papers only displaying and recording qualitative measures of performance, this section will be used to encompass results where no numeric values are recorded. One example of qualitative data is a reference to the intuitive nature of a proposed solution, which may be a subjective comment passed by the authors. Other examples may include the responses of test subjects regarding the effectiveness of a solution.

Quantitative data: Clearly the quantitative data refers to the numeric results formulated by the study of the dependant variables. Only numeric data that formed results and could be sufficiently compared to other studies was obtained. Examples include false positive rate of the algorithms.

In a number of papers, a qualitative approach is better suited if the authors are not making a direct comparison with another solution; therefore, general comments can be made on whether the solution achieved its desired aims. In essence, if a paper is reviewing the responses of security analysis using a newly developed piece of software, this is more appropriately managed with qualitative classifications and analysis, as opposed to the binary yes or no response. Naturally, in other instances, purely quantitative data, such as time required by a solution to solve a problem, or the number of vulnerabilities found, is more effectively analysed using numerical values, giving each solution a foundation from which to compare [23].

2.4 Data Analysis

There are many challenges faced when synthesising the data within this study due to the range of different algorithms, models and frameworks that are used to integrate AI in penetration testing and vulnerability assessment. These challenges are only exacerbated by the metric (independent variables) used to assess each of their performances. In some instances, the quantitative data associated with the performance indicators needed to be contextualised to be compared with other papers. It is suggested to attempt to cross-analyse each of the different models within their contexts to give insight, as appose to ignoring the model because it does not fit into that comparison criteria [21]. It is therefore important to identify the threats of cross-analysis between different contexts within this research area.

In additional to the collation of the quantitative and qualitative, a meta-analysis was performed so that AI techniques could be compared in their application to penetration testing and vulnerability assessment. To compare the entire range of different algorithms with their respect performance, a qualitative overview was taken based on the contextual standing of all the studies within the dataset. Aspects such as performance, efficiency at finding shortest path, and training period are just some of the comparison components being analysed. In addition, the reputation standing of the papers will be brought into question where necessary to properly assess the credibility of the study.

As a side note, none of the algorithms or applications were replicated with a means of confirming the validity of the studies, this was outside of the scope for this systematic literature review.

3. Results and Discussion

After applying the range of database queries across all the academic databases a total of 663 papers were found. A large quantity of these papers was duplicated due to the nature of some of the more holistic databases like Google Scholar and Web of Science which call upon other databases within their queries. After removing the duplicated a total of 214 unique papers remained. Following the removal of duplicate papers, the inclusion and exclusion criteria was applied to the abstract and title of each of the papers within the remaining dataset. Proceeding this criterion, the total number of papers was brought down to 17. To gain more papers relevant to the research area, backwards and forwards snowballing was utilised. After snowballing was applied an additional 14 relevant studies were found. These papers were then read in full to assess their relevance to the study, again using the inclusion/exclusion criteria (fortunately all papers selected fitted this criterion). 31 total relevant studies were found. It should be noted that this small dataset of papers is due to this emerging area of leveraging AI in penetration testing and vulnerability assessment. The lack of any relevant literature reviews in this area also supports this observation. We also note that the stringent and comprehensive nature of the inclusion, exclusion and quality assessment criteria may have also excluded papers with some relevance to the subject.

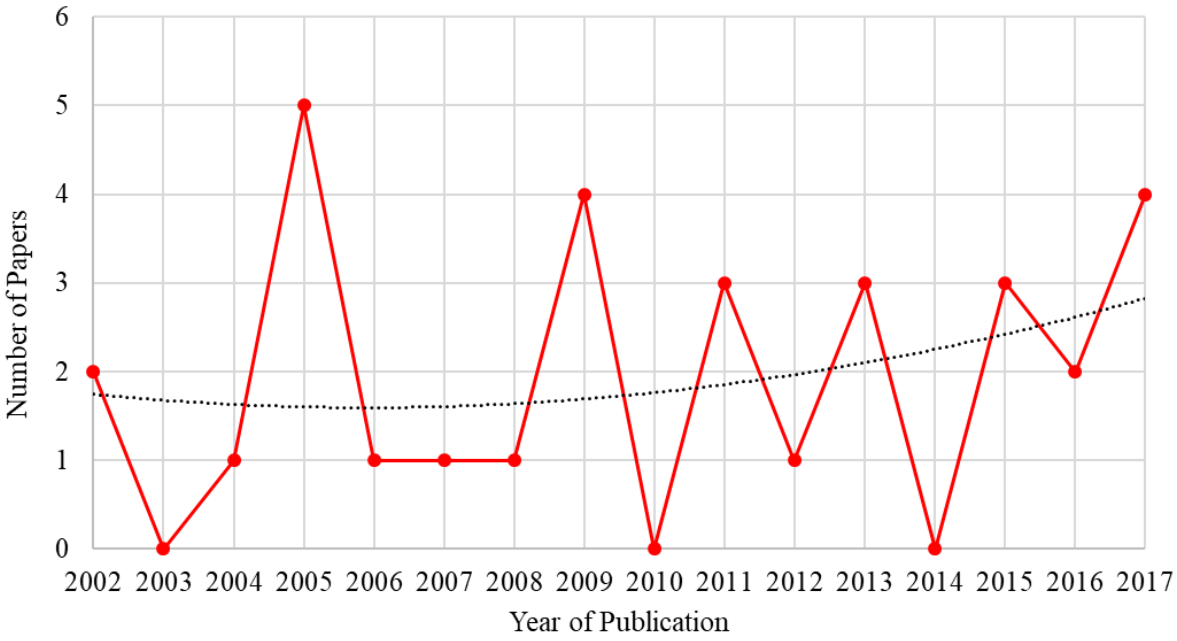


Figure 1: The rate of interest toward AI in penetration testing and vulnerability assessment.

Each of the papers was quantified respective of the year they were published in, to develop an understanding of the trends that has occurred over the recent period with regards to this research area. Figure 1 outlines the recorded tally of papers for each year of publication with the addition of a trendline to emphasise the distribution rate over the period, subsequently showing the rate of interest for this area of research. The trendline outlines a gradual rise in recent years regarding the interest of AI in penetration testing and vulnerability assessment, thus showing that interest is slowly increasing. This is reflected once again by the percentage of papers (51.61% - 16 papers) that has been published in the current decade (2010 onwards). Interestingly, out of the 31 total papers, 21 papers (67.74%) are conference papers, with the remaining 10 papers (32.26%) being peered reviewed journal papers. These results are indicative of the maturity of the research area, showing that area is still very much in its infancy relative to more established research topics.

A large number of papers incorporated the use of partially observable markov decision process (*POMDP*) and/or planning domain definition language (*PDDL*), which suggested that these attributes are a particularly applicable area of application within this field of study. Due to the prevalence of these attributes within this research area, attention will be paid to them during the modelling techniques section of this analysis (*Section 3.2.*). The distribution of the different areas of AI within penetration testing and vulnerability assessment is exhibited in the heatmap below (Table 3). This heatmap is a generalisation of each of the areas, as naturally all the papers have developed their own solutions that vary slightly in their composition.

Table 3: Heatmap distribution of AI disciplines within Penetration testing and vulnerability assessment.

Knowledge Base and Inference Engine	1
Contingent planning using PDDL	9
Genetic Programming	4
Unsupervised Learning	1
MDP (POMDP) (Reinforcement Learning)	9
Analia (Consensus)	1
Attack Simulation	2
Attack Graphs / Trees	6
OVAL	1
Genertia Red Teams (GRT)	2

Due to the restriction on the number of referenes (no more than 25 for this journal), we refer interested reader to our Github link[1] the research papers and calculations carried out in this paper.

3.1 Performance comparison in relation to Independent Variables (IV).

A range of independent variables (*IV*) are used to assess the proficiency of proposed systems within the evaluated research area. Under the consideration that some systems are somewhat analogous with others not being applicably comparable this section will compare only those papers with similar metrics, however it should be noted that despite the differences in approaches, all papers selected have a similar objective, this objective being; furthering the research area of penetration testing and vulnerability assessment with the use of AI methods. As summarised in Table 4, the range of metrics used to assess the proficiency of an AI application within the penetration testing domain varies greatly. Despite the range of metrics, most of papers within the dataset chose very unequivocal means of testing their proposed systems, thus independent variables were very succinct leading to a more identifiable objective for the dependent variables to aspire to. Many of the studies within the dataset use common independent variables, however studies differ in their measurement of the dependant variable and therefore cannot be compared in the same context.

[1] https://github.com/deanuniversityofsalford/Systematic-Literature-Review---AI-in-Penetration-testing-and-vulnerability-assessment.

Table 4: Studies and their respective metrics (Independent Variables (IVs)

Type of Metrics (Independent Variables)	**Studies (see footnote 1)**
Problem Size (*also outlined by an increased size difficult under Plan Generation Step*)	[S5], [S17], [S21], [S26]
Number of hosts in exposure (*measure of how long it takes for a sample size to be compromised or solved in relation to exploitation steps*).	[S9], [S10], [S14], [S15], [S18], [S22], [S23], [S24]
Genetic generation (*by measure of generation and effective outcome*).	[S3], [S27], [S28], [S29], [S30]
Training Epoch	[S12]
Network State (*Hosts, Vulnerabilities and software*) or Network State (*Device type*)	[S1], [S2], [S4], [S8], [S11]
Number of objectives	[S31]
Action Model	[S13], [S16]
AI engine	[S6]
Connectivity	[S7]
Vulnerabilities	[S19], [S20], [S25]

A small handful of studies ([S5], [S17], [S21], [S26]) use a common IV to assess the performance of their proposed AI intelligence implementation, this IV being the size (network) of the problem presented to the AI solution (*note*: the "*generation step*" within studies [S5] & [S17] refer to an increased size difficulty with regards to testing). Figure 2 outlines the results of each of these papers, highlighting the performance respectively. Fortuitously, each of these studies also uses the logarithmic base 10 axis measurement to show the exponential growth in the dependant variable (in this case time). This undoubtedly shows that the higher the complexity of the problem (*more complicated vulnerable network*) given to the AI solution, the more complex and time consuming the solution must be to solve the problem (essentially finding exploits for each of the units on the network). For the most part these four studies show similar results in terms of solving the problem of build efficient attack graphs applicable to their problems.

Interestingly, both [S5] and [S17] supplement their processes with the liberal use of PDDL or *Planning Domain Definition Language*, however the application is used in entirely different ways, thus showing that studies using plan generation in conjunction with PDDL do not directly perform better than those that don't with regards to speed of generation. The outcome, as seen in Figure 2, show a rapid acceleration of action planning in [S5], before levelling out to somewhat mimic the results of [S17]. This evidently shows that the cFF or Contingent Fast Forward planning process of FIDIUS within [S5] is superior in application to that of LPG-td (Local search for Planning Graphs) planner of [S17].

[S26] uses a pruning technique to optimise the decision-making process of the algorithm itself, making the decision points of each of the attack graphs easier to compute. This evidently has aided in the construction of optimised attack graphs, especially when compared to the results of [S21]. It can be assumed that although the results are only recorded up until a problem size (network size) of 4, the existing results can be extrapolated to inevitably show that [S21] performs exponentially worse than [S26]. This inherently shows that optimisation is key when working with exponentially increasing metrics or in this case "*problem sizes*".

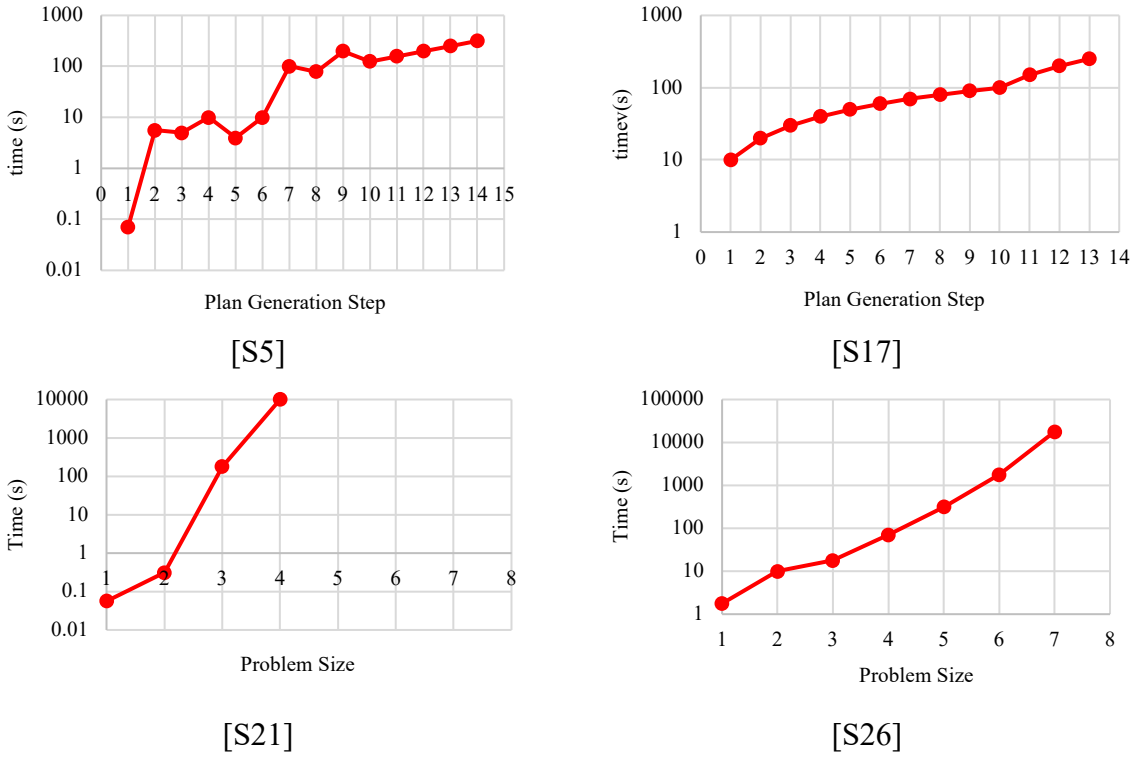


Figure 2: Showing comparable results with studies that use problem size as an IV.

Another small set of studies ([S3], [S27], [S28] & [S30]) take an entirely different approach to the sentient application of software to solve problems regarding penetration testing and vulnerability assessment. These studies facilitate the use of evolution to achieve the most applicable means of determining the best plan of action to take to compromise the proposed system(s). Much like most genetic algorithms, fitness is the metric used to determine how well suited an outcome is to the process at hand. In the studies described above fitness is defined in various ways relevant to the independent variable. Both [S27] and [S28] for instance use the process of presenting the genetic algorithm with malicious packets so that the system "learns" of the requirements needed to avoid detection within the system (avoidance level is unequivocal to that of fitness level). As previously mentioned, the fitness of studies [S3] and [S30] is defined in a different way despite the overall concept of fitness being relative to how effective an outcome is at solving the problem at hand. Studies [S3] and [S30] both define fitness as a means of developing strategies to best solve the exploitation of a system within the environment given.

The way in which the application of these models differs with regards to the independent variables is that [S3] and [S30] describe the independent variable as a generation evolution i.e. the variant of code within the solution is the determining factor of its fitness (dependent variable), as appose to [S27] and [S28] in which it is the variation of data presented to the algorithm which facilitates the measure of the fitness.

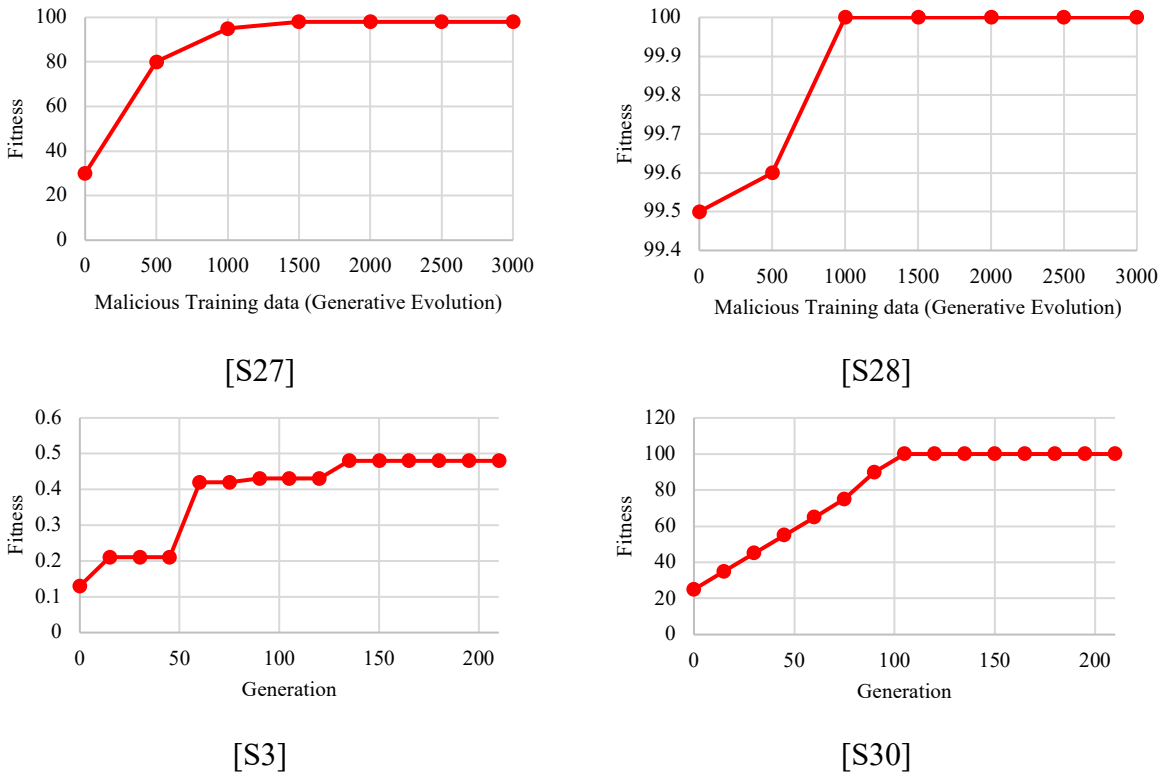
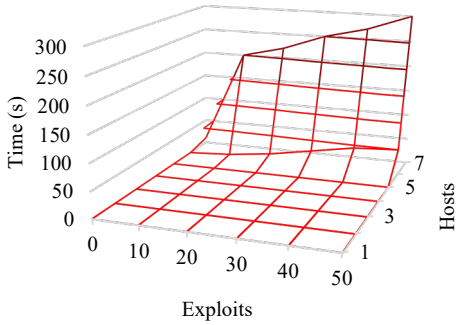


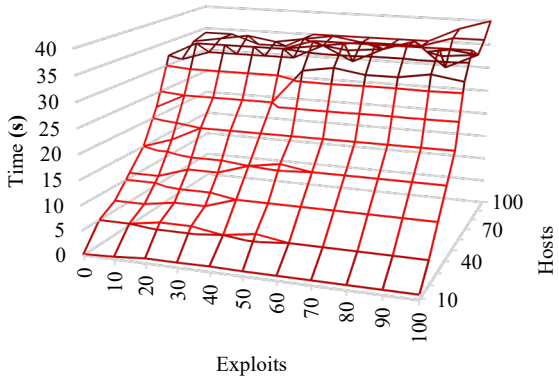
Figure 3: Showing comparable results with genetic studies that comparable IVs.

[S29] also made use of fitness as a DV in relation to generation as an IV, however a mean fitness level was taken after a multitude of tests, thus this could not be compared graphically with its respective literary peers. The independent variable associated with [S29] is comparable to the generation variable of both [S3] and [S30], in which new generations of solutions are put forward for testing and evolved depending on the fitness level reached by the solution, thus changing the independent variable being measured. In all 3 studies ([S3], [S29] & [S30]), the independent variable of generation is synonymous of the length of the solution put forward. Using generation length appears to be a desirable metric when producing a gradual development of fitness, in contrast to generative evolution which appears to spike rapidly, then levelling out on a plateau with no evident signs of improvement.

Contrary to the other studies, [S9] & [S10] make use of two independent variables as a means of testing the attack planning efficiency of their POMDP models. [S9] shows promising results using small-scale metrics for the independent variables, showing an increase in time exertion towards a higher host and exploit quantity. [S9] concludes by raising the issue of scaling as a new problem that limits POMDP in its application to penetration testing and vulnerability assessment.



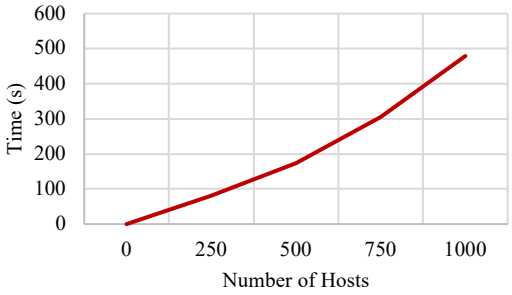
[S9]



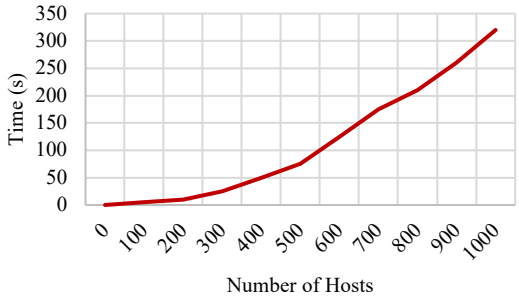
[S10]

Figure 4: Showing comparable results for studies that utilise multiple IVs.

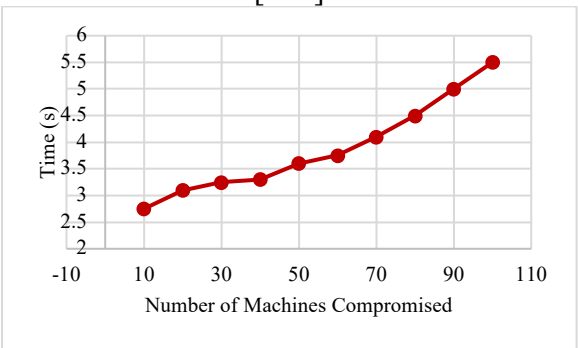
The subject of scalability is also prevalent in [S10], as [S10] shows that the application of POMDP does not scale well with the consolidated addition of the proposed 4AL model in respect to the extension of the independent variables. However, as outlined by the authors of this study, the scaling issue may be a result of encapsulation the entire attack solution for the dataset into a single POMDP. Therefore, these results show that when testing POMDPs within a penetration testing or vulnerability assessment capacity, that scalability should be taken into consideration, with independent variables extending to a large capacity to test the AI component to its maximum potential. [S6] does in fact use multiple IVs; the network type, the goal of the exploitation and the engine being used. [S6] is not able to be compared with [S9] and [S10], due to the small number of hosts on the test networks and difference in goals.



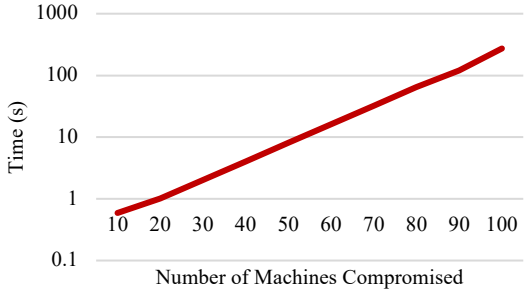
[S15]



[S22]



[S24]



[S23]

Figure 5: Studies within similar metrics based on machine exposure size.

Studies ([S15], [S22], [S23] and [S24]) all use a similar metric when testing their proposed tools and models. These studies use the quantity of machines exposed to the tool/model as a base for comparison against the quantity of time needed to overcome each specified quantity of machines. [S15] and [S22] appear to have similar results respective of their IVs, despite using entirely different models in application of AI. However, the performance of [S24] is far more effective at compromising machines in the comparison of these independent variables, despite the different in size with regards to the independent variable. As a means of further comparing these the effectiveness of this study, this IV should be extended to reflect the other studies. It should be noted that in this comparison [S23] uses a logarithmic Base 10 scale to captivate the exponential rise in time needed to compromise each unit of ten machines. This scale in conjunction with the small ranged metric of machines demonstrates that the AI application within this study is by far the most ineffective of those with similar metrics. Other similar metrics relating to the exposure size mentioned above include that of [S31], in which the authors use the IV of the number of exploit objectives required in each of the tests done, measuring the time taken to find a solution for these objectives.

S8 abstractly relates to these metrics also, however, specific network configurations were used as an independent variable as appose to the size of the hosts within a network (although this may have been applicable somewhat applicable, due to the size of each of the respective test networks in this study). This abstract relation is also true of study [S14], in which the number of hosts is the independent variable, however it is the coverage and number of exploit that are discovered for each host (as appose to exploitation) that is under measurement.

A large quantity of the studies ([S20], [25], [19], [11], [2], [1], [16], [18], [13]) use their own test networks (bespoke networks, tailored for their own needs) to assess the performance of their proposed solution. The fundamental problem related to this is that it is difficult to gauge the exact difficulty of the test networks used in each of the samples and therefore comparisons between different studies prove difficult. For example; what constitutes as an easy network in one study may not be so easily solved by another algorithm, therefore making the results subject only to the model proposed in that individual study.

Study [S12] utilises the training period as its independent variable, with the exposure to the training period being utilised to test the damage that may be caused by this. Surprisingly, this is one of the only studies that actively uses a training period as the influencing variable within the dataset despite the strong connotations of learning and training in AI. Respectively, this may be because it is one of few studies within the dataset that actively produces DDoS attacks as a means of testing the network itself.

3.2. Analysis of Modelling techniques.

The application of AI within penetration testing and vulnerability assessment manifests itself in a variety of different forms. For the most part, many models appear to encompass some degree of attack planning, be it through attack graph generation, attack tree modelling or other forms of planning. Table 5 illustrates the different models of each of the studies, with each study taxonomized based on a general overview of its respective model.

Table 5: Studies and their respective models.

Technique	Studies
MulVAL	[S21], [S24]
NuSMV (Model Checker)	[S19], [S20]
POMDP (Partially Observed Markov Decision Process.)	[S2], [S6], [S8], [S9], [S10], [S13], [S14], [S16], [S26]
MIPSA (Mixed Initiative Planning and Scheduling Agent).	[S25]
FF (+ cFF) Planner	[S5], [S22], [S23], [S31]
SGPlan (planner)	[S18],
Genetic evolution	[S3], [S27], [S28], [S29], [S30]
Reinforcement Learning (RL)	[S12]
Host Clustering	[S4], [S7]
Multiple Models	[S1], [S11], [S15], [S17]

Most of studies that focus on attack graph generation and exploitation preplanning use two predominant AI methods to facilitate this functionality. These being the Markov Decision Process (MDP) in some capacity (often in a partially observable study in which the user can influence some aspects of the output - POMDP) and genetic algorithms that generate solutions through iterations of generational fitness. This reflects that these AI techniques may be accepted solutions within the research circles of this domain. In addition to these core techniques, [S5], [S22], [S23] and [S31] all make use of the fast-forward (FF) with some making use of the contingent fast-forward (cFF) model to complement their application. All models using these processes to generate attack plans outperform any other model within their comparison group. All papers conclude that the use of fast-forward planning allows for the creation of shorter attack paths with contingent options being supplemented if required (with a longer lead time naturally).

Other studies, such as [S26] use SPUDD (Stochastic Planning using Decision Diagrams) in conjunction with MDP (mentioned above), this combination outperforms UID (unconstrainted influence diagrams) in [S21] (which makes use of MulVAL, the main area of research for [S24]), with both models translating attack graphs into their respective models. UID does appear to scale better than those models which use MDP based algorithms, as a slower rise (not exponential like that of MDP) in the required time is noted over the increasing size of the network. Unfortunately, due to the network size tested, it is difficult to determine if the UID model maintains a stable solution in terms of scalability.

For the most part, models that use MDP all suffer from scalability issues with regards to attack graphs. One prime example of this is [S9], in which a knowledge base is used in conjunction with POMDP to generate exploits for a small number of hosts. As the host number gets higher, towards end of the study the time needed to find exploits for these devices grows exponentially. [S10] (using the same base model), tries to remediate the scalability issue using an additional 4AL Decomposition Algorithm to refine and create policies for each of the attack graph scenarios. This does not work however, as the issue of scalability is not resolved, only made worse. [S14] is another example, however in this study the authors limit the model to a 30-minute testing time, ensuring that the attack graph does not grow out of control. This does not fix the issue of scalability however. Other studies like [S26], make use of pruning techniques to increase the efficiency of attack graph generation. [S26] also makes a point of simulating a real-world environment in which the attack graphs it

creates take it to account the interaction with honey pots (no other attack graph generation utilise this sort of sensibility).

Other studies make use of MDP to develop attack graphs and/or trees to enable the construction of adversary profiles. Studies like that of [S6] use classical planning through a composition of MDP and Monte Carlo simulations to develop and build attack graphs. However, due to small dataset tested by this study, it is difficult to determine if this bespoke algorithm suffers from the issue of scalability. Due to the centralised use of MDP to develop the algorithm it could be extrapolated that this is certainly the case.

In contrast to MDP models that use classical planning, [S8] advocates the use of contingent planning over classic planning, concluding the use of contingent planning (essentially sensing outcomes that could potentially be, as oppose to marking all potential actions as a viable path) is predominately more efficient. [S22] uses a bespoke planning algorithm based off a range of other algorithms, incorporating PDDL consolidation to ensure that the issue of scaling in attack graph creation is properly mitigated against, which inevitably worked to an extent but requires more testing to develop a comprehensive conclusion regarding the model.

Other studies that address the performance issues of POMDP generation models, are that of [S2], which structures its objectives on the refinement of the POMDP based approaches to enable the scalability and general performance of these models to be effective in real world environments. [S2] proposes a range of different optimisation criteria including budget optimisation and fault tolerances, again progress is made in optimising the system, but nothing solid is concluded. The authors do however highlight the need for real network testing to enable them to draw conclusions as to the increase in efficiency of their proposed optimisation.

Two studies ([S20] and [S19]) both make use of the NuSMV (model checking) which both comprehensively check attack graph generation to determine the viability of the proposed graph. Once an attack graph is generated [S20] analyses the results to determine the most reliable path available. This process, as denoted by the authors is extremely time consuming and does not scale well. Conversely, [S19] shortens attack chains to the shortest path to ensure that exponential explosion (scalability issue) does not occur with regards to decision branches within the graphs themselves. The time-consuming overhead of [S20] may also be a result of the model generation and mapping of the initial network to generate the foundation for attack graph.

Many of the studies make use of the PDDL (Planning Domain Definition Language) to consolidate and standardise the action needed for attack planning. [S17] makes use of PDDL as a means of translating common vulnerabilities and attack platforms into structured attack plans. PDDL appears to be the most widely used standardised language for vulnerability assessment in this domain. Contrary to the use of PDDL as a planning language for building attack graphs and other applications, some models use OVAL (Open Vulnerability Assessment Language). For instance, Study [S14] states the declarative descriptions of OVAL to be better in their application of creating more centralised knowledge base around certain exploits and therefore providing a better means for analysis in comparison to PDDL. [S24] also makes use of OVAL, underlining that it is an excellent tool for gathering information regarding host configuration, but later stating that it is only a basis for information gathering in this respect and needs further refinement.

As previously mentioned during the IV section, [S5] outperformed [S17] with this performance potentially being attributed to [S5]'s use of the PDDL planning language within a knowledge base. This is true for most of the models that incorporate PDDL. [S25] uses PDDL translation to provide a more granular and descriptive personalised attack graph that allows for a more user specific attack graph to be created, this may not be so easily developed with the use of OVAL, as OVAL gives general descriptions regarding exploitations. The use of PDDL within [S25] affirms the fact that models that use PDDL perform significantly better than those that do not, despite not using MDP. In study [S18], the use of PDDL aids in maintaining the shortest path possible for minimal attack graph generation, thus aiding with the issue of scalability.

Stepping away from the use of attack planning, a slightly more unusual AI paradigm in the form of genetic evolution algorithms occurs frequently within the dataset of this study. Both [S27] and [S28] both make use of GENERTIA Red Teams (GRTs) as a genetic model for the foundation of their overall model. In contrast [S3] and [S30] make use of grammatical evolution, in which the script or solution is grammatically altered, thus a new generation is 'born'. These two groups of genetic studies use different metrics to assess the efficiency of their solutions, so it is difficult to assess the performance in relation to each other. However, for the most part, all the genetic algorithms within this study perform similarly in that they improve significantly over the same generation span, thus showing that no evolutionary model is more effective than another in the application of penetration testing and vulnerability assessment.

3.3. Analysis of Evaluation criteria

The methods in which the performance is measured within the AI in penetration testing and vulnerability assessment domain differs greatly. Table 6 shows the full extent of the different evaluation techniques used by the studies within the paper. As mentioned in the analysis of the Independent Variables, it is difficult to completely compare studies that are similar due to the extent of combinations between the IV's and the evaluation criteria.

Table 6: Studies with their respective evaluation methods.

Evaluation Criteria	**Study**
Time (host + vulnerability discovery)	[S15], [S24]
Time (number of compromises)	[S5], [S17], [S21], [S22], [S23], [S26],
Host / Exploit coverage	[S14]
Fitness (generational)	[S3], [S29], [S30]
Fitness (based on learning model)	[S27], [S28]
Damage	[S12]
Attack steps generated	[S19], [S20], [S25],
Minimal number of steps generated	[S18]
Level of uncertainty	[S13]
Cluster Deviation	[S7]
Probability of clustered devices	[S4]
Viability of Solution	[S1], [S2], [S8], [S11], [S16], [S31],
Multiple evaluation criteria	[S6], [S9], [S10],

As illustrated in Table 6, a large quantity of papers ([S5], [S9], [S10], [S15], [S17], [S21], [S22], [S23], [S24], [S26], [S31]) purport to evaluate their models using time as a

performance metric. The time evaluation method is often associated with a model's ability to quickly enumerate or exploit a host or number of hosts. Study [S14] follows a similar approach to the large quantity of studies that use time to measure performance of enumeration or exploitation of a large network. However, differences occur within this study as it does not dwell on the time efficiency required to exploit, enumerate or find vulnerabilities on the hosts. Instead it focuses on a more fundamental value surrounding the total coverage of vulnerabilities on each of the hosts within the dataset, taking a more qualitative approach. This study effectively shows that the number of hosts considerably affects the model's ability to find suitable exploitation means to take control of targets. This is surely a more imperative goal of a vulnerability assessment agent, than the time that a service takes to enumerate a small set of obvious vulnerabilities.

Upon analysis both studies [S8] and [S31] both have very specific objectives or tasks to perform when testing on specific hosts or networks, with both models performing considerably well on all the tasks given. In addition to other metrics such as the number of actions generated for their proposed solution, both utilise time as an evaluation method. In reference to their times in comparison to other models, both have exceedingly fast times for enumeration of exploits for their chosen hosts or networks. It is hard to distinguish whether the fast times are due to the use of the Fast-Forward metric model (mentioned previously) or due to their concentrated efforts on specific scenarios. If the latter is the predominant cause of the increased performance, it is perhaps a better and more fortuitous venture to focus research efforts in the future on more streamlined and distinct objectives.

Peculiarly, the evaluation method of [S12] is the damage caused, this is summarised by the number of malicious packets being processed by the network. The authors state that the primary function of this research was to show that the learning algorithm could improve over time, and that the actual recording of the assessment criteria (damage) was not a base for comparison. This bespoke evaluation criteria makes comparing this algorithm to other learning-based algorithms inherently difficult. Nonetheless, this assessment criteria does offer some benefits when evaluating penetration testing and vulnerability assessment models, as assessing the overall damage (or increase of damage overtime) may be a more befitting method of assessing the overall outcome (or damage) that a potential breach in security may cause.

A range of studies ([S6], [S9], [S10]) use multiple evaluation criteria to assess the productivity of their solution. The multiple assessment criteria associated with [S6] allow for a more granular assessment of the AI solution. Not only is the time to enumerate the hosts recorded, but later stages of the testing such as the number of accounts enumerated and the amount of data exfiltrated also included. This multiple stage of recordings allows the authors to test the model at multiple stages of the exploitation stage to determine how efficient this model is at performing suitable penetration tests. Going forward, testing models at multiple stages of the penetration testing scenario may offer better insight into the successes and failings of future models on a more granular level.

4. Meta-Analysis

Due to the range of metrics and results put forward by models using AI for penetration testing and vulnerability assessment, a meta-analysis is needed to combine these studies in a shared context, with the addition of appropriately weighting each of the studies to determine the overall agreeable conclusion on application of AI in penetration testing and vulnerability

assessment. This meta-analysis unlike many others will contain mostly qualitative studying of each of the papers. For the most part it is difficult to incorporate all the studies into one analysis as there is no standardised basis for testing i.e. every study uses a different network to test their solution. To avoid bias during this meta-analysis, all papers, even those that are not easily comparable, will be analysed to ensure that the overall picture is given. In an addition effort to avoid bias, each paper's citations (where it is applicable) will be analysed to determine the renown of the paper under the premise that papers with high citations will be more acceptable as a means for comparison and application than those with a low count.

To begin, the productivity of [S12] increased by 60% over the training period of the study, showing that the training period respectively had a great influence on determining the effectiveness of the solution. This paper is the only paper that utilises reinforcement learning (MDP based) with a training epoch to solve the problem, thus showing that MDP based attack map generation may not be the most productive solution for penetration and vulnerability assessment in terms of AI. The credibility of this paper is brought into question by the low number of citations (10), despite the paper having a multitude of results. The use of reinforcement learning with the addition of the damage report evaluation metric may be more applicable to the AI application in this domain, the MDP based attack graph generation.
Models that perform well are that of [S5], which uses Fast-Forward planner (+FF). The results for this model are astounding with multiple layers of testing showing reliable results and a respectively small decrease in efficiency over larger networks. In addition, this model makes use of both the PDDL and MSF framework (an already consolidated and well-known exploit knowledge base), which may have aided in its success.
Study [S31] uses both PDDL and Fast-Forward planning (+FF), which inevitably lead to extraordinary results, the source of the results may be attributed to the combination of these two characteristics. The fame of this paper pays homage to its achievements with it being cited a total of 99 times. [S23] also uses both PDDL and Fast-Forward planning (+FF), again, with outstanding results in comparison to those with MDP/POMDP models. Following on the theme of citations this paper has also be cited 42 times. [S18] uses *SGPlan,* which is a variant of Fast-Forward planning (+FF). This study was found to be slightly more efficient than its predecessors.

The NuSMV checker model in [S20] is one of the keystone publications with regards to automated generation of attack graphs, having been cited 1,321 times, despite this model is highly time consuming. This study gave a good basis for other research in the AI in penetration testing and vulnerability assessment domain. A potential contender of MDP is study [21], using UID (a combination of MulVAL and OVAL language). The performance of the problem solver was extremely good using this combination of attributes.

In [S6], the authors provide a more realistic basis of AI application in penetration testing and vulnerability assessment, stepping away from the conventional exploit search of other studies, looking at a more realistic set of actions (i.e. account compromise and data exfiltration). However, despite multiple tests being carried out, the overall sample size analysed in this study does not give conclusive results regarding the overall application of AI in this research area and should therefore be weighted as less influential than other more detailed studies, due to being unable to determine if scaling is an issue in this real-world application technique. [S24] is one of the few models using the OVAL vulnerability language and the MulVAL scanner in comparison to those models that use MDP and PDDL, this model is seen as one of the most promising application of intelligent agents and planning

within this domain, this is again reflected in the number of times cited (459). A crossover occurs within [S14], with this study favouring to use OVAL over PDDL within an MDP model. Naturally, scalability issues still ensue, and the overall performance of the model suffers when the number of hosts increases. [S22] uses a bespoke algorithm with the PDDL language, which unfortunately does not perform as well as models that use MDP with PDDL (This is also the case for [S15]).

As mentioned previously the scalability of POMDP is a giant issue, especially for studies like that of [S10] and [S9]. Many papers have tried to mitigate this issue using various techniques. [S26] for instance uses pruning to ensure that MDP does not become unscalable, this aids in MDP becoming a useful application for large penetration tests, even coming close to rivalling the previously discussed MulVAL model. The optimisation efforts of [S2] also allow this model to be a contending against MulVAL, however more testing is required with regards to this study. CycSec in [S1] may also prove worthy, but the level of testing is limited to small networks and cannot be a yet viable solution. In contrast to MDP studies, [S25] using MIPSA makes great progress with regards to refining the graph development process, this study also uses PDDL.

Finally, genetic algorithms form a small group within the overall group of AI in penetration testing and vulnerability assessment. Study [S3], improves fitness by almost 500%, with [S30]'s fitness improves by over 300% (this is also true for [S27]). [S28] only improves by just under 1 %, but it should be noted that this solution is already running at a near 100% fitness value.

4. Potential Research Directions

There have also been attempts to achieve scalability to some extent, for example using fast-forward (+FF) and contingent fast-forward (+cFF) planning to facilitate attack graph planning. However, striking a balance between scalability and effectiveness remains one research opportunity, particularly as systems become more complex.

A number of AI models use detection time as a quantifying metric, for example, in terms of seconds or minutes. One potential research direction is to design AI-based approaches to identify vulnerabilities that are exploitable, and to what degree, in real-time.

To achieve optimal performance, we recommend that future solutions should probably be designed for the application domain or specific system, rather than been overly abstract or generic. Abstract applications, such as those designed to search for or generate general exploits over many hosts, may only detect simple or on-the-surface type vulnerabilities. Also, future AI models should test multiple stages of their models during the different penetration testing stages (e.g., initial compromise compared with post exploitation), which is likely to facilitate more realistic simulation and learning.

We also observed that most studies use testbeds or simulations for evaluation, rather than real-world systems. Hence, one potential research opportunity is to design a suite of standardised systems / networks that are sufficiently realistic, complex and have features that are typical of different real-world applications. This will provide researchers a common platform to evaluate and benchmark their approaches (this is analogous to using a common dataset to benchmark the various performance metrics of some security approaches).

Assessment criteria should also probably move away from time towards more qualitative measures, for example "consequence / impact" as a means of determining the effectiveness of AI applications (or any other) in penetration testing or vulnerability assessment models.

5. Conclusion

While the utility of AI in penetration testing and vulnerability assessment has been demonstrated in existing studies, we need to keep pace with technological advances and evolution of adversarial techniques to avoid detection. In other words, the application of AI in vulnerability assessment is expected to be a growing area, and its role and importance will increase as systems become more complex in our smart and connected society.

Based on our review, we made a number of observations, such as those reported in Section 4. For example, one observation made in this paper is that scalability is a key challenge, for example in approaches based on POMDP attack graphs and genetic algorithms. Genetic algorithms appear to perform better over time in relation to their independent variables, due to the nature of generation evolution (since genetic evolution allows one to evolve the required solution to a point where it is adept to its environment). This also suggests that the application of general AI techniques needs to be carefully considered in the context of the application environment, in order to have the best "fitness" required to complete the task at hand. Hence, future solutions / approaches should incorporate some elements of evolution to produce the best suited or "fittest" solution possible.

References

[1] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955," *AI Mag.*, vol. 27, no. 4, p. 12, 2006.

[2] E. Mjolsness and D. DeCoste, "Machine Learning for Science: State of the Art and Future Prospects," *Science (80-.).*, vol. 293, no. 5537, pp. 2051–2055, 2001.

[3] J. Hendler, "Avoiding another AI winter," *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 2–4, 2008.

[4] T. O. Ayodele, *Types of Machine Learning Algorithms*. InTech, 2010.

[5] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, 2010.

[6] C. E. Brodley, U. Rebbapragada, K. Small, and B. Wallace, "Challenges and Opportunities in Applied Machine Learning," *AI Mag.*, vol. 33, no. 1, pp. 11–24, 2012.

[7] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, 2018.

[8] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. Tutorials*, vol. PP, no. 99, p. 1, 2015.

[9] N. Milosevic, A. Dehghantanha, and K. K. R. Choo, "Machine learning aided Android

malware classification," *Comput. Electr. Eng.*, vol. 61, pp. 266–274, 2017.

[10] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence," *IEEE Trans. Emerg. Top. Comput.*, 2017.

[11] N. A. Naik, G. D. Kurundkar, S. D. Khamitkar, and N. V Kalyankar, "Penetration Testing: A Roadmap to Network Security," *J. Comput.*, vol. 1, no. 1, pp. 187–190, 2009.

[12] M. Petraityte, A. Dehghantanha, and G. Epiphaniou, *A model for android and iOS applications risk calculation: CVSS analysis and enhancement using case-control studies*, vol. 70. 2018.

[13] J. Yeo, "Using penetration testing to enhance your company's security," *Comput. Fraud Secur.*, vol. 2013, no. 4, pp. 17–20, 2013.

[14] D. Geer and J. Harthorne, "Penetration testing: A duet," in *Proceedings - Annual Computer Security Applications Conference, ACSAC*, 2002, vol. 2002–Janua, pp. 185–195.

[15] BAE Systems, "Intelligence-Led Penetration Testing Services," 2015.

[16] S. Doğan, A. Betin-Can, and V. Garousi, "Web application testing: A systematic literature review," *J. Syst. Softw.*, 2014.

[17] I. Hydara, A. B. M. Sultan, H. Zulzalil, and N. Admodisastro, "Current state of research on cross-site scripting (XSS) - A systematic literature review," *Information and Software Technology*. 2015.

[18] M. Rouse, "What is vulnerability assessment? - Definition from WhatIs.com," 2018. [Online]. Available: https://searchsecurity.techtarget.com/definition/vulnerability-assessment-vulnerability-analysis. [Accessed: 19-Sep-2018].

[19] M. Rouse, "What is penetration testing? - Definition from WhatIs.com," 2011. [Online]. Available: https://searchsoftwarequality.techtarget.com/definition/penetration-testing. [Accessed: 19-Sep-2018].

[20] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," *Engineering*, vol. 2, p. 1051, 2007.

[21] S. Hosseini, B. Turhan, and D. Gunarathna, "A Systematic Literature Review and Meta-Analysis on Cross Project Defect Prediction," *IEEE Transactions on Software Engineering*, 2017.

[22] Oxford Living Dictionaries, "Definition of artificial intelligence in English by Oxford Dictionaries," 2018. [Online]. Available: https://en.oxforddictionaries.com/definition/artificial_intelligence. [Accessed: 19-Sep-2018].

[23] M. Walker, "Quantitative vs Qualitative Research: What, Why, Where, When and How to Use Each." [Online]. Available: https://www.askattest.com/blog/home/quantitative-vs-qualitative-research-and-how-to-use-eachcdr. [Accessed: 19-Sep-2018].