PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 2 [August, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

AI RECOMMENDATION PENETRATION TESTING TOOL FOR CROSS-SITE SCRIPTING: SUPPORT VECTOR MACHINE ALGORITHM

Nur Saadah Salim
2021462508@student.uitm.edu.my

Shahadan Saad*
shahadan@uitm.edu.my

Article Info

Received: August 2024
Accepted: March 2025
Available Online: August 2025

Abstract

This research introduces a new approach to enhancing cybersecurity by integrating Support Vector Machine (SVM) algorithms with penetration testing to develop a recommendation system focused on Cross-Site Scripting (XSS) attack detection. By leveraging AI and machine learning, the system dynamically suggests the most suitable penetration testing tools which are Nmap, XSStrike, PwnXSS, OWASP ZAP, or Burp Suite. The SVM algorithm, a supervised learning model, plays a crucial role in improving the efficiency of tool selection, ultimately enhancing the speed and adaptability of vulnerability detection processes. The research employs Extreme Programming (XP) methodologies to ensure high-quality code, agility, and collaborative development. The methodology focusing on simplicity and fast development process that suitable for simple and small project in a short time. The system is developed using Django as the main framework for Python on an Ubuntu server with SQLite as the database, utilizing a dataset synthesized in Microsoft Excel and aligned with real-world examples from Kaggle. Extensive testing, including unit, integration, and acceptance testing, has validated the system's functionality, although limitations such as a focus on XSS testing tools and some accuracy concerns were identified. This study contributes to the cybersecurity domain by offering a scalable, AI-driven solution that integrates traditional penetration testing practices with advanced machine learning techniques. Future work will explore expanding the system to include automated reporting, integration of ChatGPT, and support for a broader range of attack vectors, addressing current limitations and further enhancing the system's utility.

Keywords: Pentest, AI, SVM and XSS

INTRODUCTION

Penetration testing is a crucial cybersecurity practice that simulates cyberattacks to identify vulnerabilities in systems and networks, ultimately enhancing an organization's security posture (Sushmita, 2021). Traditionally, penetration testing was a manual and labor-intensive process, but advancements in technology have led to its automation, significantly

improving speed and accuracy. In particular, the integration of Artificial Intelligence (AI) and Machine Learning (ML) has revolutionized penetration testing by enabling more efficient and effective vulnerability detection.

AI, especially through machine learning techniques, has become an indispensable tool across various industries, including cybersecurity. By 2035, AI is expected to contribute $15.7 trillion to the global economy, underscoring its transformative potential (Duggal, 2020). Machine learning, a subset of AI, utilizes algorithms to identify patterns and relationships in data, making predictions and generating insights that can enhance security measures (Tucci, 2023).

In the context of penetration testing, AI-driven tools can analyze vast amounts of data, adapt to emerging threats, and provide recommendations for appropriate testing tools, thereby improving the efficiency and effectiveness of security assessments (Bisen, 2019). This project aims to develop an AI-based recommendation system for selecting penetration testing tools focused on Cross-Site Scripting (XSS) vulnerabilities. The Support Vector Machine (SVM) algorithm, known for its proficiency in binary classification, is employed to analyze user inputs and suggest tools such as Nmap, PwnXSS, and XSStrike, which are specifically designed to detect XSS attacks. The suggested tools consists of Nmap, PwnXSS, XSStrike, OWASP ZAP and Burp Suite.

The system is developed using the Django as Python framework on an Ubuntu server, with SQLite as the database. By implementing the SVM algorithm, the system can accurately match user requirements with the most suitable penetration testing tools, reducing the time and effort needed for manual tool selection. The ultimate goal is to create a scalable, AI-driven solution that simplifies the penetration testing process, enhances security professionals' productivity, and strengthens web applications against cyber threats.

This research not only contributes to the development of AI-enhanced penetration testing tools but also provides valuable insights into best practices for integrating AI into cybersecurity, potentially making the internet a safer environment for both users and organizations.

LITERATURE REVIEW

Artificial intelligence (AI) is changing security testing by emphasizing the use of deep learning and machine learning algorithms to mimic attack scenarios and strengthen defences. The chosen Python framework is Django with SQLite as database is a good choice as it is related and a good framework to work with XP methodology. With the use of SVM, we attempt to create an AI recommendation penetration testing tool that will find the best tool depending on user requirements more precisely. The review emphasizes the SVM algorithm's versatility and the need for further study to increase its effectiveness in vulnerability identification. The literature research demonstrates that this project can assist the tester and enhance vulnerability assessment.

Penetration Testing

Penetration testing is an important security procedure that finds and fixes vulnerabilities in network infrastructure. It is a key component in protecting companies from online attacks. As an alternative to cybercrimes, this technique mimics actual attacks to find vulnerabilities in network servers and operating systems (Saber et al., 2023). Network, web, mobile apps, and cloud systems represent only a few of the areas that penetration testing covers to properly identify and manage vulnerabilities in each of these areas (Drolet, 2022).

Manual and automated testing are the two main methods used in penetration testing. Although manual testing is time-consuming and expensive, it permits a complete and detailed review when conducted by qualified engineers. On the other hand, automated testing, which is powered by pre-written scripts and algorithms, is quicker and more effective, but it may produce more false positives and less accuracy (Saber et al., 2023). Although automation makes testing large systems more efficient, manual testing is still important because it might find complex weaknesses that automated methods would overlook. To accomplish a thorough security assessment, a hybrid strategy that combines automated and manual procedures is frequently utilized (Hance et al., 2022).

The amount of information supplied prior to engagement has a major influence on the results of a penetration test. Organizations select from a variety of penetration test types, including white box, black box, and grey box testing, to define the scope whether broad or narrow, set priorities, and efficiently manage expenses and time. Different amounts of information are disclosed to the tester both prior to and during the assessment in each category.

White Box

When conducting White Box testing, the penetration tester is fully aware of the system and software, including the IP address that governs each component (Garg & Bansal, 2021). Full-knowledge testing is another name for it, in which the attacker has access to target information prior to the test. This approach examines an application's internal workings and functioning, testing each internal component through the source code. White Box testers must be skilled programmers who have a thorough understanding of how the application functions. Assuming a developer profile, the tester differs from Black Box testing. Finding second-order vulnerabilities in online applications that need two access or time attack inputs is a critical function of White Box testing (Saber et al., 2023). After the initial entry, the second input could result in an error or allow access to compromised data. This helps stop vulnerabilities from being exploited, however before the second attack input is entered, the online application must already have a security weakness. By restricting the attack vector to two inputs, its efficacy is increased.

Black Box

Also known as functional testing, black box testing occurs when the tester is not aware of the target beforehand (Garg & Bansal, 2021). Attackers use fingerprinting techniques to learn as much as possible about the target without having to know how it operates internally. This testing technique, which mimics the activities of actual hackers, necessitates extensive, in-depth investigation prior to the pen test. This approach is time-consuming and expensive, even if it is effective in discovering current system vulnerabilities because it mimics a real cyber-attack by a black hat hacker. The pen tester is capable of simulating attacks from both the inside of the system by an authorized user of a web application and the outside by a user who has been granted access with the intention of performing malicious actions (Saber et al., 2023).

Grey Box

Grey box testing involves providing a limited quantity of information to an attacker who has little knowledge with IP addresses and URLs (Garg & Bansal, 2021). This method frequently involves a pen tester evaluating the extent of insiders' access within the targeted network. The same privileges are granted to both the attacking team and regular users,

mimicking an insider attack with malicious intent. The attacker is given more information about the target, including network configurations, subnetworks, or particular IP addresses, in this last stage of the penetration testing procedure, which expedites the reconnaissance stage (Saber et al., 2023). Grey Box testers, in contrast to Black Box testers, possess a basic comprehension of the device's operation against which the assault would be initiated. This knowledge may include design and architectural documentation, as well as a network account, making the testing process more efficient.

Cross-site Scripting (XSS)

The widespread usage of cross-site scripting (XSS) assaults in phishing, information theft, and other malicious actions is a result of their covert implementation techniques and wide targeting. XSS vulnerabilities arise when there is no strong security against malicious code and happen during the creation of a web application (Guan et al., 2022). Malicious scripts are injected into a victim's browser when they visit an infected website or application by hackers in an XSS attack. Anyone who visits the hacked website can then receive these scripts (Guan et al., 2022). Particularly vulnerable to XSS attacks is unclean user input, which gives hackers the ability to take cookies and pretend to be users (En & Selvarajah, 2022,). Because the compromised webpage can be sent without any obvious indication of an assault, XSS prevention is difficult.

XSS attacks, which include both stored and reflected XSS, are highly dangerous for the security of personal data and enterprises. They divide into three main categories which are DOM-based XSS, reflected XSS, and stored XSS (En & Selvarajah, 2022,). There are two forms of server-side cross-site scripting attacks consist of persistent and reflected (Sethi et al., 2023). Reflected attacks produce unencoded output. In stored cross-site scripting scripting (XSS), payloads are injected as user input, stored on a server, and then performed when a victim accesses the website. While DOM-Based XSS is more difficult to identify since it takes advantage of client-side data flow, Reflected XSS tricks victims into requesting harmful data (Sethi et al., 2023).

Artificial Intelligence (AI)

Artificial intelligence (AI) has become an effective technology that is transforming several industries in recent years. One area where AI is having a big impact is penetration

testing, which is essential to guarantee the security of networks and computer systems (Kambo, 2023). Additionally, cybersecurity has rapidly emerged as a grand societal challenge of the 21st century and it makes technology experts find a way to secure the technology, device or system (Samtani et al., 2020). AI is a popular option in cybersecurity due to its capacity to identify patterns and analyze massive amounts of data. This explanation demonstrates why artificial intelligence (AI) is a fantastic tool for quickly evaluating massive amounts of incident- related data, modeling social engineering attacks, detecting real assaults, and identifying vulnerabilities.

AI is transforming security testing by improving productivity and quickly recognizing vulnerabilities, doing complex and unique tests, and removing the need for time-consuming human labor—all of which make it simpler to discover potential vulnerabilities (Küçükkarakurt, 2023). AI has been developed to detect and prevent human error, enhancing cybersecurity by detecting code vulnerabilities and proactively addressing them to prevent irreversible security issues (Küçükkarakurt, 2023). AI is transforming penetration testing by helping businesses find vulnerabilities and improve their defenses. AI solutions use machine learning and deep learning algorithms to simulate attack scenarios, helping businesses stay ahead of cyber adversaries (Kambo, 2023).

Machine Learning

Machine learning, a branch of computer science and AI, focuses on recognizing patterns and making predictions. It is categorized into four main types which are supervised, unsupervised, semi-supervised, and reinforcement learning. Supervised learning uses labeled data to build models, while unsupervised learning uncovers hidden structures in unlabeled data. Semi-supervised learning combines both labeled and unlabeled data, and reinforcement learning involves a software agent optimizing actions to maximize rewards (Thomas & Gupta, 2020).

The machine learning process typically involves seven steps which are data collection, pre-processing, feature engineering, training, testing, performance evaluation, and making predictions (Farhat et al., 2020). Data is gathered, cleaned, and organized before being used to train models. These models are then tested and evaluated using metrics like accuracy and precision. Ultimately, machine learning enables computers to perform tasks based on learned data, making it a powerful tool for prediction and analysis across various fields.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 2 [August, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

Naïve Bayes Algorithm

The Naïve Bayes algorithm uses the concept of probability to classify data by totalling up the frequency and combination of values from the dataset either independently or in dependence on other data to determine the class of an object whose label is unknown (Fadil et al., 2022). The formula of naïve Bayes algorithm: $P = (C \mid X) = \frac{P(X|C)P(c)}{P(x)}$. X represent data class that is known or will be searched, c is probability of data P(c) is probability of hypothetical data and P(x) is the probability of the c value (Fadil et al., 2022). The advantages of Naïve-Bayes include its ability to work with binary data, its fast and efficient calculation, its ability to ignore missing or empty data, and the absence of need for a large amount of training data. A few drawbacks include the inability of the condition's probability value to be zero, the need for many probability measurements to assess the degree of accuracy, the capacity to only identify words and the reliance on preliminary data for success (Fadil et al., 2022).

K-Nearest Neighbors Algorithm

K-Nearest neighbours (kNN) algorithm classifies a new data point based on majority class among its k-nearest neighbours (Sitawarin & Wagner, 2020). Like all gradient-based attacks, kNN is fast and efficient, and it provides a reliable baseline for evaluating kNN-based models. In particular, KNN does not require a training period and makes use of pre-existing data for predictions (Sitawarin & Wagner, 2020). Its implementation is made simple by calculating distances between data points using formulas like Manhattan or Euclidean, which allows for the seamless integration of new data and ensures flexibility and continuous adaptation in predictive tasks (Soni, 2020). However, KNN has difficulty calculating distances between every data instance in a large dataset due to the computational cost of doing so, and high-dimensional data exacerbates this problem by making distance calculations more complex (Soni, 2020). The decision function for kNN is

$f(x) = argmax_{c \in C} \sum_{i \in N_k(x)} I(y_i = c)$. C is the set of all possible classes, $N_k(x)$ is the set of indices of the KNN neighbors of x, I is the indicator function, which is 1 if y_i=c and 0 otherwise while y_i is the label of the i-th nearest neighbor.

Support Vector Machine Algorithm

A key algorithm in the statistical learning approach, support vector machine (SVM) is a traditional supervised machine learning algorithm that is primarily utilized in binary classification problems (Le et al., 2019). SVM is a controlled AI model that addresses problems based on innovative information characterization and repeatability (Kumar et al., 2022). Furthermore, SVM is a crucial part of AI machine learning technology that is used in conjunction with text mining methods to look at the timeliness of news and the relationship between news and oil price (Mohamed & Messaadia, 2023). Based on a few articles about SVM, the difference between the SVM use is in Table 2.3. This comparison aims to evaluate if cloud-based penetration testing using the SVM technique is feasible.

The SVM algorithm aims to create the best decision boundary in n-dimensional space, known as a hyperplane, by selecting extreme points or vectors called support vectors. This decision boundary helps categorize new data points in the correct categories, making it easier to categorize new data points in the future. Decision equation of Support Vector Machine algorithm: $f(x) = sign(w \cdot x + b)$. The bias term is b, and w is the normal vector to the hyperlane. The SVM can be trained to determine the values of w and b. A fresh data point called "x" cannot be classified until the values of w and b have been determined.

Decision Tree Algorithm

Decision tree algorithms are a popular machine learning technique. It is used to classify data based on its simplicity and automatic control. The core technology in data mining for this algorithm is a decision tree that represents a classification or decision set based on features. It generates development rules and regulations using information retrieval as a basis for attribute segmentation. Decision Trees have different algorithms used to build trees, such as ID3, CART and C4.5.

Iterative Dichotomiser 3 (ID3) is a decision tree algorithm that uses acquired information as a basis for attribute segmentation, rule generation and rule development for classification or decision sets (Jiang, 2023). ID3 is a prime example of data mining and machine learning. Algorithm C4.5 is based on ID3, focusing on feature attributes for data classification based on information acquisition rate. In addition, Classification and Regression Trees Algorithm (CART) is a probability theory and statistical approach (Jiang, 2023). CART is used in decision tree analysis to handle skewed or polymorphic numerical data and ordered or

unordered class attribute data. It uses the GINI coefficient for classification tree nodes and the minimum sample variance for regression tree nodes.

Random Forest Algorithm

A random forest algorithm is created by many decision trees for training and shows the results in the classification prediction of each tree. The random forest algorithm is a predictor consisting of properties based on tree random regression. Random forest classifiers offer similar hyperparameters but can be used independently (Donges, 2021). They add randomness to the model by finding the best feature among a random subset of features, resulting in a wide variety and better model. Additionally, the Random Forest Algorithm is a powerful tool for quantifying the relative importance of each feature on the prediction. Sklearn calculates by comparing the impurity reduction across all tree nodes using the feature (Donges, 2021). These scores are automatically calculated after training and scaled to ensure that all interests are equal.

METHODOLOGY

Methodology used is Extreme Programming (XP) which an agile project management methodology that emphasizes speed, simplicity, and adaptability through short development cycles and minimal documentation. It is particularly suited for small, customer-focused teams with strong technical expertise. In XP, user stories are used to capture requirements from the end-user perspective, guiding feature prioritization and acceptance testing. An architectural spike is conducted to reduce uncertainty and understand potential challenges before full development begins. Release planning involves determining the scope, prioritizing features, and setting timelines for incremental releases, allowing for continuous refinement and adaptation. Spikes are also used to explore potential solutions or resolve issues, often resulting in code that is discarded after serving its purpose. Software development in XP follows iterative cycles, with each iteration producing a potentially shippable product increment and involving planning, design, testing, and integration. Acceptance testing is performed collaboratively with customers to ensure that features meet specified criteria and align with business requirements. XP emphasizes delivering small, frequent releases to the customer.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 2 [August, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

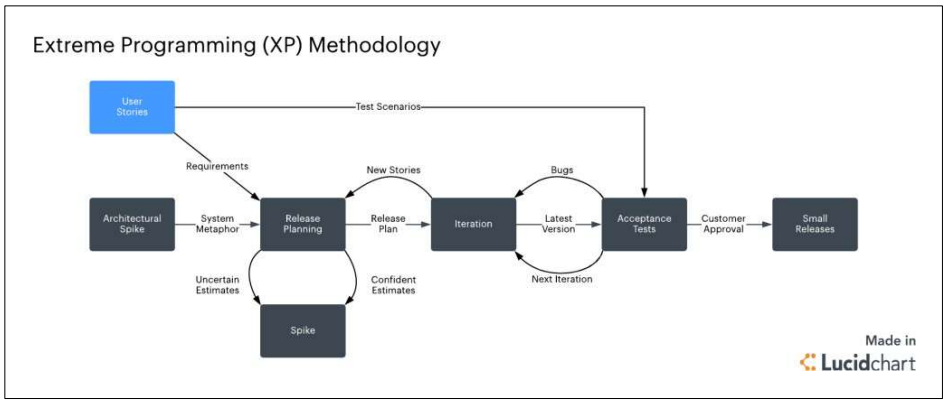

Figure 1: Extreme Programming Methodology

User Flowchart

In this system, user need to sign in into their account if they have one and register if does not have account yet. For register, user need to enter their username, password and email address. To request for penetration testing tools suggestion, user need to enter the required requirement. After the suggestion appear they need to give feedback either they agree with the suggestion provided or not. If user does not agree, they can give their own suggestion. The feedback given by user will be saved into database. After that, the user can chose to run the tools suggested by the system or to see user's history for their earlier suggestions. To run the tools, user need to enter the target URL or IP address according to the tools requirements. User also can see their past tools running result.

System Architecture

Using Django, the system's backend handles database interactions that using SQLite, data processing, and essential logic, all while maintaining security and dependability. The backend consists of multiple layer which are business layer, AI model layer, Pentest Tools, database and data layer. The backend manages form submissions, runs machine learning models, obtains and stores data from the database, and manages user authentication. Additionally, it uses tools like Nmap, PwnXSS, and XSStrike to do network scans and interfaces with external APIs. A user-friendly interface with multiple pages is provided by the frontend, which was constructed with HTML, CSS, and JavaScript. On the other hand, the client side manages user interactions via web browsers and delivers updates and feedback in real-time. The smooth and dynamic experience across several platforms is ensured by the design of both sides, which are compatible with a wide range of devices and browsers.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 2 [August, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

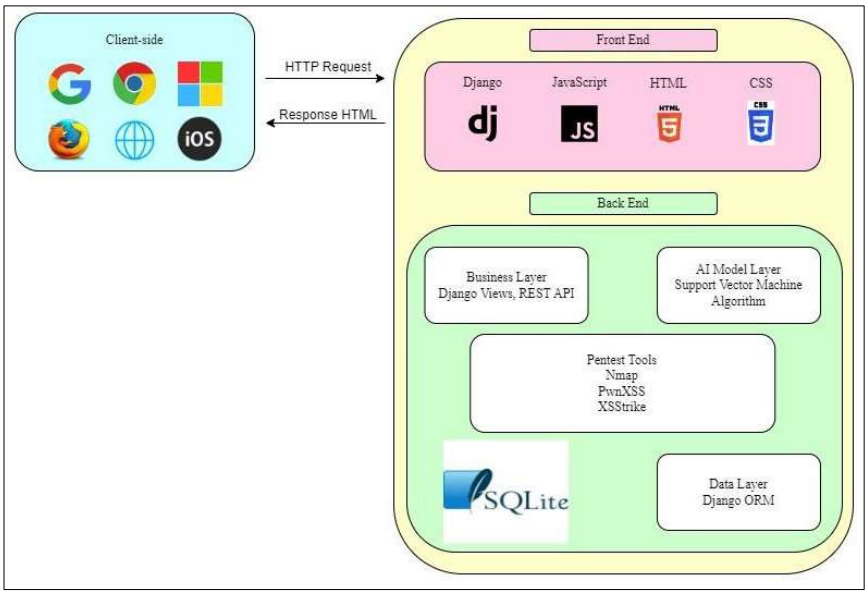

Figure 2: System Architecture

RESULT AND DISCUSSION

To develop the AI system for this project, the Support Vector Machine (SVM) algorithm was selected due to its effectiveness in classification tasks. The dataset was preprocessed by converting all entries into numerical form, facilitating faster and more efficient data training. The scikit-learn library, a robust Python tool for machine learning, was used to implement the model. Specifically, the SVC class, a specialized implementation of SVM for classification tasks, was utilized. In the training process, the feature matrix (X) included attributes such as goals, attack type, skill level, automation, and platform, while the target (Y) represented the tool suggestions based on user requirements. The model's performance was evaluated by testing it on 20% of the dataset, with accuracy results providing insights into its effectiveness. Once training was complete, the trained model was saved using joblib for reuse within the system. In this research 3 tests had been did which are unit, integration and user acceptance test. Unit testing ensures software functionality in isolation, while integration testing ensures smooth integration of modules. The Django-developed backend successfully interacts with the web interface, handling data processing, user authentication, and the execution of tools like Nmap, PwnXSS, and XSStrike. It demonstrates robustness and reliability of this system. Acceptance testing verifies the functionality of the system from the user's perspective, providing accurate

tool recommendations. Successful unit testing, integration testing and acceptance testing demonstrate the system's reliability and readiness for use.

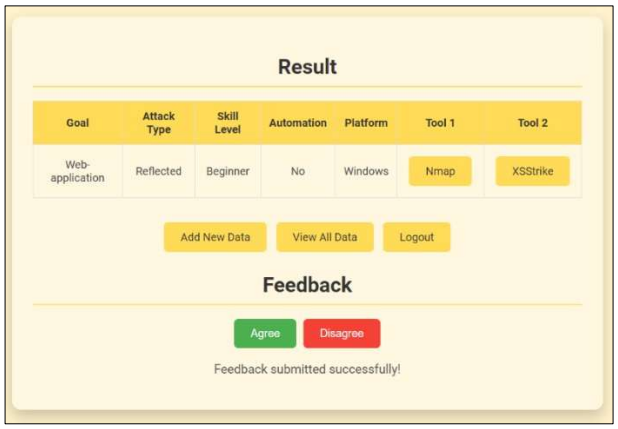

Figure 3: User Feedback Result

CONCLUSION

This project has several limitations, including the need for a more extensive and comprehensive dataset. While the current dataset, created with Microsoft Excel and supplemented with Kaggle examples, provides a foundation for testing, a larger dataset would improve the accuracy and reliability of the tool's suggestions. The accuracy of the algorithm's recommendations is also limited by the quality of the dataset and the complexity of modeling user needs for penetration testing tools. Only three tools which are Nmap, PwnXSS, and XSStrike had implemented, while OWASP ZAP and Burp Suite are suggested but not integrated, limiting the system's coverage of available tools. Future work will focus on enhancing the system with automated reporting, expanding the range of tools, and incorporating interactive features similar to ChatGPT. These improvements will increase the system's utility, making it a more comprehensive and user-friendly tool for penetration testing. Overall, the project successfully demonstrates the potential of using AI to streamline penetration testing, offering a valuable resource for cybersecurity professionals.

REFERENCES (APA 7TH EDITION)

Donges, N. (2021, July 22). Random Forest: A Complete Guide for Machine Learning. Built In. https://builtin.com/data-science/random-forest-algorithm

Duggal, N. (2020, April 22). What is artificial intelligence and Why it matters in 2024? Simplilearn.com. https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/what-is-artificial-intelligence

Fadil, I., Helmiawan, M. A., Supriadi, F., Saeppani, A., Sofiyan, Y., & Guntara, A. (2022). Waste Classifier Using Naive Bayes Algorithm. *2022 10th International Conference on Cyber and IT Service Management (CITSM).* https://doi.org/10.1109/citsm56380.2022.9935894

Farhat, R., Mourali, Y., Jemni, M., & Ezzedine, H. (2020). An Overview of Machine Learning Technologies and Their Use in E-learning. *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*. https://doi.org/10.1109/octa49274.2020.9151758

Garg, D., & Bansal, N. (2021). A Systematic Review on Penetration Testing. *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 1-4. https://doi.org/10.1109/gcat52182.2021.9587771

Guan, H., Li, D., Li, H., & Zhao, M. (2022). A Crawler-based Vulnerability Detection Method for Cross-site Scripting Attacks. *2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS- C)*. https://doi.org/10.1109/qrs-c57518.2022.00103

Hance, J., Milbrath, J., Ross, N., & Straub, J. (2022). Distributed Attack Deployment Capability for Modern Automated Penetration Testing. Computers, 11(33), 1-19. https://doi.org/10.3390/computers11030033

Jiang, W. (2023). Design of Body Function Index Mobile Query System for College Students Based on Decision Tree Algorithm. 12th IEEE International Conference on Communication Systems and Network Technologies, 539–544. https://doi.org/10.1109/csnt57126.2023.10134611

Kambo, M. (2023, May 23). *The Transformative Impact of AI on Penetration Testing*. LinkedIn. https://www.linkedin.com/pulse/transformative-impact-ai-penetration-testing-mary-kambo

Küçükkarakurt, F. (2023, April 1). *Is It Possible to Use Artificial Intelligence for Penetration Tests?* MUO. https://www.makeuseof.com/is-it-possible-to-use-artificial-intelligence-for-penetration-tests/

Kumar, K. V., Malathi, P., & Ramesh, S. (2022). Performance Analysis of Placement Prediction System using Support Vector Machine over Random Forest Algorithm. *2022 14th International Conference on Mathematics, Actuarial Science,* Computer Science and Statistics (MACS). https://doi.org/10.1109/macs56771.2022.10023271

Le, W., Lin, M., Jia, L., Ai, J., Fu, X., & Chen, Z. (2019). Multi-Objective Optimization of an Air-Cored Axial Flux Permanent Magnet Synchronous Machine with Segmented PMs based on Support Vector Machine and Genetic Algorithm. *2019 22nd International*

Conference on Electrical Machines and Systems (ICEMS). https://doi.org/10.1109/icems.2019.8922465

Mohamed, N. A., & Messaadia, M. (2023). Artificial Intelligence Techniques for the Forecasting of Crude Oil Price: A Literature Review. *2023 International Conference On Cyber Management And Engineering (CyMaEn)*, 340-343. https://doi.org/10.1109/cymaen57228.2023.10050945

Saber, V., ElSayad, D., Bahaa-Eldin, A. M., & Fayed, Z. (2023). Automated Penetration Testing, A Systematic Review. 2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 373-380. https://doi.org/10.1109/miucc58832.2023.10278377

Samtani, S., Kantarcioglu, M., & Chen, H. (2020). Trailblazing The Artificial Intelligence for Cybersecurity Discipline. *ACM Transactions on Management Information Systems, 11*(4), 1-19. https://doi.org/10.1145/3430360

Sethi, M., Verma, J., Snehi, M., Baggan, V., Virender, & Chhabra, G. (2023). Web Server Security Solution for Detecting Cross-site Scripting Attacks in Real-time Using Deep Learning. *2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON 1)*. https://doi.org/10.1109/icaia57370.2023.10169255

Sitawarin, C., & Wagner, D. (2020). Minimum-norm Adversarial Examples on KNN and KNN Based Models. *2020 IEEE Security and Privacy Workshops (SPW)*, 34-40. https://doi.org/10.1109/spw50608.2020.00023

Soni, A. (2020, July 3). *Advantages and disadvantages of KNN*. Medium. https://medium.com/@anuuz.soni/advantages-and-disadvantages-of-knn-ee06599b9336

Thomas, R. N., & Gupta, R. (2020). A Survey on Machine Learning Approaches and Its Techniques:. *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*. https://doi.org/10.1109/sceecs48394.2020.190

Tucci, L. (2023, September 15). What is machine learning and how does it work? in-depth guide. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML