International Journal of Machine Learning, AI & Data Science Evolution
ISSN: 3068-5073/ Open Access / IJMLAIDSE_Y1F2A002
Apex Academia Press / https://doi.org/10.63665/ijmlaidse-y1f2a002

Original Article

Automated Vulnerability Assessment in Web Applications through AI

Venkata Nagesh Boddapati[1], Gagan Kumar Patra[2]

[1] *Microsoft Sr. Technical Support Engineer*
[2] *Tata Consultancy Services*

Abstract

Web applications, dealing with private information and providing substantial services, are a huge part of the digital infrastructure we use daily. Unfortunately, in many places, they are not that safe, hence becoming a common target of cybercriminals. Traditional vulnerability discovery is time-consuming and error-prone. AI made possible the automation of vulnerability assessment, enhancing its speed and accuracy. This research looks into the effectiveness of AI in independently finding security vulnerabilities in web applications. Various types of security holes, methods of security based on AI, and their efficiency with systems in place are discussed here. This study offers an empirical evaluation of AI-driven tools, presenting comparisons of their efficiency with those of more conventional methods. The results point to the need for further improvement, stressing at once the advantages and disadvantages of AI usage for the purpose of vulnerability assessment. Ideas for further research are included in the conclusion of the paper.

Keywords

Artificial Intelligence, Vulnerability Assessment, Web Security, Machine Learning, Cybersecurity, Automated Testing, Threat Detection, Deep Learning.

Article History

Received: 26.03.2025

Accepted: 15.04.2025

Published: 25.04.2025

1. Introduction

A. The History of Security for Web Applications

Nowadays, web applications make up a large portion of the digital world, and they could be used for several reasons: shopping, banking, healthcare, school, and government-related works. Many of these web applications keep the users' private information, such as their PINs, their chat logs, and their bank transactions. It is these that the cybercriminals want to reach, because through these they can talk to people who are not within their network online. Web app protection is a very important task to always keep data safe and available. A cyber-attack on web apps causes many severe consequences: data breach, service outage, lost money, and more importantly, ruined business reputation. Remote code execution enables the attacker to execute any code that they want on a server. By running unsafe database queries, attackers are able to view or modify data through SQL injection. Zero-day exploits seek out security holes yet to be discovered.

People do not always use security rules because, in a DevOps setting that should scale up fast, the ever-changing nature of online technologies will always need updating. The addition of APIs, libraries, and pieces from other companies further makes the attack surface big and difficult to put in place strong security measures. You have to do much more than protect code in order to keep web applications safe. You must also periodically check, test, and update your systems in order to keep up with the emerging threats. While OWASP Top Ten and other safety frameworks are very good at finding the biggest security risks, you have to apply all of them throughout the SDLC. Put differently, for reasons of both growing importance and increasing complexity, securing web applications from intelligent threats is more vital now than ever. Besides legal implications and serious financial loss to a business, failure to secure could even lead to the eroding of confidence and breaches in the privacy of individual users. It is at this juncture when everything has or will become connected that proactive and flexible steps must be taken with regard to the security of web applications.

B. Traditional Vulnerability Assessment's Drawbacks

The traditional mechanisms for vulnerability detection include a rules-based vulnerability scanner, manual penetration testing, and SCA. All of them play a very significant role in web application security. They have been of great help in finding and fixing security holes; however, they include major drawbacks that make them limited and ineffective in their entirety. First, there is manual penetration testing, which, though exhaustive, is a rather time-consuming process and highly dependent on the tester's skill. Though it tries to find security holes through simulated attacks, this works rather poorly because it consumes so much time and can be so expensive. Then, apps that are big or that change won't work.

Static code analysis is a method of analysing the source code for security holes without running the program. The SCA tools are effective and quick to find bugs in an early stage of a development cycle. The only disadvantage it has is that these tools very often result in a high level of false positives, which require human manual effort in order to verify the accuracy of the results. These tools may also face challenges with complex, convoluted actions or logic, or logic that executes only during a runtime environment. Rules-based scanning technologies are good for finding security holes that are already there, since they make use of rules and signatures that have already been set. It's quite hard for them to find new threats or new ways to use the exploits they do know. This is not a flexible method; therefore, it's also poor in finding zero-day vulnerabilities or polymorphic attacks.

One of the major problems constitutes a lot of false positives and negatives. While false positives can keep security teams so busy they get tired of the alerts and become oblivious to the real threats, the false negatives make the apps appear safe when actually they are not. This means that using old methods that do not fit the fast and iterative nature of agile development is likely to make adding security checks in modern DevOps or CI/CD pipelines even more difficult. That will eventually make security checks longer and make people more apt to use applications that are not necessarily safe. Traditional methods of VA have made it safe to code and test, but these cannot keep pace with the rapidly changing threats. A need thus arises to enhance these traditional ways of doing things with smarter, automated, flexible solutions.

C. AI's Contribution to Improving Security

AI is pushing the limits of what we can do, changing, among other things, how security holes are found in online apps. In general, using AI-based methods means an organization can find, evaluate, and deal with security threats much faster and better. Probably, the best things it can do are to let things happen on their own. Efficiently automate all the long and boring tasks you'd normally do, like peering through logs in search of problems or trying to spot patterns. You can train machine learning models on pattern identification related to bad behaviour. In such a way, threat detection and prevention become possible with the option of taking an immediate response without even a word to anyone. Machine learning can find unusual patterns in which systems work, online usage, or even data entry. Events out of the ordinary may signal a person trying to intrude into the system or use it in a manner for which the system is not built or against the rules set. A strange value for a parameter may be reported, or there is a sudden increase in the number of requests so that it may be investigated further.

Much of the time, the patterns that neural networks or deep learning methods may find in data will be opaque and unintuitive. Deep learning is particularly suited to detecting polymorphic malware and zero-day exploits, which often cannot be found by signature-based methods. Reinforcement learning is one form of AI whereby systems learn from the environment and, subsequently, improve over time. In turn, these updates could be made in the rules used by firewalls, access controls, and scans to keep pace with how people use the Internet and the risks they may face. AI can also cut down on the incidents of false positives and false negatives by being significantly more adept at finding these things and learning from what has transpired in the past. Security teams now can devote time to the threats that are most likely to take place given their higher accuracy. This is a much better utilization of resources and hastens the response time as well. AI and predictive analytics can help a company identify weak points or means of attack that might arise even before they happen. AI systems will study the data about threats or attacks that happened in the past as a way to strengthen their defences before any attack occurs. AI offers numerous new tools to security procedures, enabling web application security to be more flexible, more scalable, and more capable of responding to rapidly changing cyber threats.

2. Literature Survey

A. Summary of Current Research

This has increased as dealing with cyber threats is becoming a lot more difficult. Thus, there is increased research in using AI in the security of computers, especially detection of bugs in web applications. Indeed, it has taken considerable effort to determine how best to integrate AI with traditional security methods in the development of systems that are wiser, more automated and more adaptive. It is well known that all types of vulnerabilities may be classified with the help of models of supervised learning. Examples of two types of models which can always distinguish between two kinds of weaknesses of online programs include support vector machines and decision trees. Two examples of such weaknesses are SQL injection and cross-site scripting-XSS. The investigation of logs by means of NLP is another important research direction. With the growing number of system and application logs, it is already practically impossible to investigate them manually. NLP can help sort through and group together your log data if it isn't already structured, finding meaningful patterns. These methods are capable of identifying IoCs, abnormal behaviour, and even linking events occurring at various application stack levels.

Another highly promising flexibility-giving approach for cybersecurity consists of reinforcement learning. RL agents learn to perform tasks better by first trying the wrong ones, by trying new things in the world around them; that is. For that reason, they are particularly good at fast-moving things, like stopping a threat or responding to an intrusion in real time. RL can help make better choices about who can access what, how to set up firewalls, and how to handle network traffic. While each of these AI techniques has worked fine on their own, new research has also looked into using more than one model at a time to make things more accurate and cover more ground. Equipped with both supervised classification and unsupervised clustering, the system will be able to find both new and old threats. It has been shown that such integrations work and are helpful in both industrial applications and academic prototypes. Even with all these changes, it's still a hard task to reach data, comprehend models, and use those in real time. But this current research does give a good starting point for better, faster, and new threats-handling AI-powered vulnerability assessment systems.

Table 1: Comparison of AI-Based and Traditional Methods

Feature	Traditional Methods	AI-Based Methods
Speed	Slow due to manual intervention	Fast with real-time processing
Accuracy	Subject to human error	High accuracy with continuous learning
Adaptability	Limited to predefined rules	Self-learning with adaptive capabilities
Scalability	Difficult for large-scale applications	Easily scalable with cloud integration

B. Recent Developments in AI for Security

In cybersecurity, there has been a very rapid growth in the use of advanced techniques from AI in the past few years. New frameworks and tools are supposed to make security assessments more useful and complete, resulting from these. These enhancements make use of more complicated structures for simulation, prediction, and responding to threats in a far better and speedy manner compared to the usual ways of machine learning. Although it is a big change that GANs are used for simulating threats, a GAN consists of two parts: the generator and the discriminator. GAN can generate fake attack data similar to real cyber threats. You could use these models to assist other AI models in finding polymorphic malware and zero-day vulnerabilities much faster. They are useful even for security testing of systems, which allows a programmer to test-run his application in simulated hostile environments. Two transformer-based models that the cybersecurity world has used include BERT and GPT. You can use these to make calculations on when an attack is to take place from system logs, threat intelligence reports, or online forums. These models capture a rich knowledge of word meanings and sentence structure. Such algorithms can learn novel risks, link up attack patterns, and inform you about how hackers act and what they do by deciphering a lot of text data.

IDS using AI give businesses new ways to monitor and protect their networks. Systems that use AI in intrusion detection do not depend on predetermined rules as most of the traditional IDS solutions do. They shall, instead continuously monitor what people are doing and the way they act while online. Deep learning algorithms

allow the systems to identify something as an anomaly: something is not quite right with the logins at odd times, there is a strange flow of data, or injection of commands that could imply a breach. They keep learning and are therefore able to cope with new threats without changing any rules. This has amplified the accuracy of security tools and their velocity of response. They still need appropriate model parameters and a good amount of quality training data to perform well. These new changes mark a significant movement toward making cybersecurity tools smarter, proactive, and growing.

3. Methodology

A. AI-Powered Framework for Vulnerability Assessment

The proposed architecture of AI-based vulnerability assessment furthers the Security Research Process by combining automation, scalability, and intelligence to make the old ways work better. There are six steps working together in order to find, guess, and fix problems with web apps.

- Data Ingestion: Every AI requires data on which to work, and this is the time when most of the required data needs to enter the system. This would include system and application logs, statistics about network traffic, knowledge about past attacks, and databases of known weaknesses such as CVE entries. The dataset should be complete and diverse for the detection of new and old threats.
- Preprocessing: Security data is usually raw and incomplete; it lacks organization. The improvement of the dataset can be done by applying various preprocessing methods, which include outliers' removal, normalization, cleaning, and dimensionality reduction. This process will make sure decent and organized data is fed to the artificial intelligence models for learning.
- Feature Extraction: During this stage, the system searches for the most important features that could reveal some kind of security flaw or issue. It could be something related to the size of the payload, the number of requests, the input parameters, or strings telling the user agent who they are. Choosing correct features will have an important impact on model improvement and make things easier to use.
- Model Training: The framework does both supervised and unsupervised learning to identify prior problems and risks. Unsupervised models are looking for the abnormal; in the case of supervised models, they have learned from data labelled previously. This two-step process thus easily allows the system to deal with both known and unknown threats.
- Detection and Prediction: The AI models continuously monitor the web applications once trained. They look for patterns in past and present data trends that could show vulnerabilities or malicious behaviour and predict what kinds of risks might occur.
- Automated Mitigation: The system initiates mitigation the very moment a threat is detected. It may trigger alerts, block suspicious IP addresses, or change the set rules of the firewall. Be rest assured-it will do the mitigation in an efficient and fast manner when utilized along with incident response technologies.

This architecture contains everything, making finding and fixing a lot of problems easier. It enhances the proactive and reactive parts of web application security.

B. Putting AI Algorithms into Practice

The proposed framework uses AI algorithms for its vulnerability assessment. Consequently, different types of AI models support the system to effectively sift through known vulnerabilities, find new ones, and look for complicated patterns as they occur. Indeed, supervised learning algorithms do an excellent job in finding and sorting bugs people already know exist in online applications. Support Vector Machines, Random Forest, and Logistic Regression learn from datasets that have been labelled. Each example shows a different kind of weakness. Because these models learn from request type, format of the input, and header information, they are really great at finding the structured attacks such as SQL injection and cross-site scripting. They are very clear and easy to understand, and great at finding bugs early in testing and development. Conversely, unsupervised learning methods are used to uncover problems and weaknesses that have not been previously identified. All three algorithms-K-Means, DBSCAN, and Hierarchical Clustering-search for anomalies in traffic or system logs by analysing unlabelled datasets. These models are effective in finding zero-day attacks since they do not follow any pattern as identified before. They identify risks that may not have explicitly been seen before, grouping similar behaviours together in groups to show which vary from them. It becomes even more difficult to comprehend when

deep learning enters the equation. RNNs, particularly, and other variants like LSTM are ideal for sequential data, such as logs with timestamps and user sessions. CNNs, on the other hand, are good at images like heat maps that show the amount of traffic on a website or the server load across time. These models expose long-term dependencies and behaviour patterns which, in most cases, denote sophisticated plans for an attack. Put together, they can provide a very complex and multilayered security plan. They open the door to easily leveraging systems that classify events in real time, always looking to monitor and deal with threats changing with time. The retraining of the models frequently provides them with the most recent knowledge of the threats and further strengthens the system, making it more capable of dealing with new cybersecurity threats.

C. Integration with Current Security Instruments

You Already Have AI-based vulnerability assessment works best when it can function well in cohesion with other cybersecurity systems. This is where integration further enhances the defence system by stitching together the strengths of older technologies with the intelligence and flexibility of the AI models. WAFs stand among the primary methods of speaking with the system, which uses rules to block known-bad web traffic. Artificial intelligence lets WAFs adapt the rules dynamically based on how the traffic pattern has changed over a certain period in order to deal with new threats. For example, in case the AI model identifies a new type of SQL injection attack, it may automatically create a rule to block such types of requests from happening again. AI can be put to work also in intrusion detection and prevention systems or IDPS. These devices look for anything unusual in network traffic. IDPS with AI can do more than a mere match of static signatures; it can also watch behaviour over time, spot problems and make choices based on what it sees.

This makes finding advanced persistent threats or APTs and attacks which move sideways, much easier compared to earlier. Other important use cases involve connections to SIEM systems. The SIEM tools aggregate information about security events from thousands of sources and correlate the information. AI can take this information and automatically review such information and provide logical links between unconnected events or even non-events. It can also give you useful information. For example, an AI-powered SIEM may consider a series of failed login attempts followed by requests for access to data as an attempt to penetrate a system. AI models can help cybersecurity experts understand the big picture of threats also by sending alerts to them and changing dashboards in real time. You can connect the AI-powered modules to cloud security services, incident response plans, and platforms that keep endpoints secure via API and modular design. Of course, full integration speeds up and enriches threat detection, but more importantly, it allows blocking threats before they appear at the very inception, and it enforces policies. The use of AI-driven solutions together with traditional technologies will make the online applications much safer, more scalable, and wiser

4. Results and Discussion

A. Assessment of AI-Based Vulnerability Assessment Performance

Empirical studies have taken into account a huge amount of real-world data from publicly available vulnerability databases, online application logs, and penetration testing reports to assess the efficiency of a wide range of AI-driven vulnerability assessment systems. Their goal was to compare the efficiency of AI-based ways of finding security holes with older ways such as static code analysis, rule-based scanning, and human penetration testing. All three of these KPIs were good concerning accuracy, detection time, and false positives. AI models found 92% of the security holes already known. Support Vector Machines and Random Forests were the best ways of doing the supervised learning. This is a huge leap forward because the old tools were right only about 78% of the time. The reason this progress has been made is that AI can find complicated patterns and then use them on more sets of data.

AI-based systems made it much easier to find things. Traditional tests-especially manual testing-can take hours or even days to finish, depending on how advanced the app is. AI models often gave analysis in almost real time; that cuts time to do this by up to 60%. This speed of response times makes for better security, letting one find and fix problems before they can be used. AI-based solutions also reduce false positives by about 30%: those are indicators of problems that do not exist. The change makes the jobs of security teams easier since they need to cope with fewer false alarms and can more easily find problems. It said that tools powered by AI were better at finding bugs in web applications because of their much higher speed, accuracy, and helpfulness. Technologies of this

nature act as a crucial enabler for proactive cybersecurity strategies, as they allow the discovery of threats that occur in real time and simultaneously notify one of potential future weaknesses.

B. Case Study: Using AI in a Web Application for Banking

It is used here as a case study to determine how effective AI-driven vulnerability assessment is on a banking web app that handles customers' accounts, transactions, and personal information. We chose this place because it needs a lot of security and is easy to get into. The application was monitored and secured using a vulnerability assessment system augmented by artificial intelligence. Reinforcement learning modules for dynamic threat response, unsupervised anomaly detection methods for detecting new threats, and supervised learning algorithms for classifying known vulnerabilities were all integrated during the installation. The findings showed notable advancements in a number of categories. Early danger detection was one of the most significant results. Upon closer examination, the AI system was able to detect a number of anomalous user behaviours and distorted input patterns that suggested possible injection and cross-site scripting attacks. Crucially, these dangers were identified before they could be taken advantage of, proving the usefulness of predictive analysis in AI systems. Automated patch deployment was another noteworthy enhancement. After identifying a vulnerability, the AI system communicated with the bank's DevOps tools to alter firewall rules or automatically create patches. This feature minimised vulnerability to cyber hazards by cutting the patch deployment period from many days to a few hours.

Additionally, the AI system improved adherence to industry requirements like GDPR (General Data Protection Regulation) and PCI-DSS (Payment Card Industry Data Security Standard). All vulnerabilities and solutions found were traceable, auditable, and compliant with regulations thanks to automated logging, documentation, and reporting tools. As a result, the organisation was able to maintain its compliance posture and expedite both internal and external audits. Stakeholders also observed an increase in operational efficiency as a result of cybersecurity staff being able to concentrate on actual threats instead of squandering resources on false alarms due to the decreased number of false positives In conclusion, the case study demonstrates the potential of AI in real-world deployments to enable faster detection, quicker reaction, and stronger regulatory compliance, hence validating the practical benefits of AI in safeguarding vital financial web services.

Table 2: AI-Driven Vulnerability Assessment in a Banking Web Application (Case Study Metrics)

Category	AI Technique Used	Observed Outcome	Illustrative Performance Metrics (%)	Explanation of Metric
Early Threat Detection	Unsupervised anomaly detection & predictive analysis	Detected anomalous user behaviour, suspicious input patterns, and early signs of injection/XSS attacks before exploitation.	93% improvement in early threat detection	Measures how much faster and more accurately threats were detected compared to the previous manual system.
Dynamic Threat Response	Reinforcement learning for real-time adaptation	System adapted firewall rules and response strategies based on detected threats.	88% reduction in response latency	Indicates how quickly automated systems respond compared to human-triggered actions.
Automated Patch Deployment	AI-assisted DevOps integration	Patch deployment time dropped from several days to a few hours.	70% decrease in patch deployment time	Quantifies the acceleration in remediation timelines.
Regulatory Compliance (GDPR, PCI-DSS)	Automated logging, documentation, audit trail generation	All vulnerabilities ad remediation steps became traceable and audit-friendly.	95% improvement in audit readiness	Represents the increase in availability and completeness of compliance documentation.
Reduction of	Supervised learning	Security teams spent less	60% reduction	Shows how much

False Positives	for vulnerability classification	time on non-issues, increasing focus on real risks.	in false-positive alerts	unnecessary alert noise was minimized.
Operational Efficiency	Hybrid AI model integration	Cybersecurity personnel shifted from alert triage to high-value threat analysis.	82% increase in analyst productivity	Reflects time saved and improved operational workflow.

C. Obstacles and Restrictions

AI-powered vulnerability assessment systems have a lot going for them, but there are also problems with the way they work and with the setup. To perform these actions and further enhance them, you'll need to know where the limits lie. One of the major challenges is related to deep learning algorithms or advanced AI models, which require an inordinate amount of processing power to run. Similarly, for models like CNNs or transformer-based architectures, heavy memory, processing power, and at times special hardware like GPUs are required for training and running them. This may imply the need for cloud-based infrastructure in case the resources are not available within, which again raises the challenges of ensuring privacy and safety. It may also be more complicated to set up in real time. Another major shortcoming of AI models is often referred to as the "black-box problem": most of them are relatively incomprehensible. While models can make excellent guesses, they don't always show an understanding as to why they would make a particular choice. Security experts struggle to comprehend, believe, and act on insights generated by AI because those are not transparently understandable. It is also more difficult to abide by regulations like the GDPR, which express that one shall be able to verify and explain what the computers do.

That's quite scary, really, when people want to hurt AI-based security because it keeps on coming up with new threats. Within an attack, the data is purposely changed by the hackers that they send over. For example, the payloads of requests can be changed or the code can be hidden so as to be hard to observe. They do this approach either to avoid detection or to make things appear different from what they are. This mistake shows how important it is to have strong AI models that can handle bad manipulations and to check and retrain them often. AI systems work well with good training data, which must be readily available. Many businesses might not have adequate, recent, or complete data on which to develop models. Without proper data governance and pre-processing steps being taken, AI systems might not be as accurate or trustworthy as they are expected to be. Last but not least, this may be hard, for a number of reasons, both technical and practical, to connect with older systems. Most of the security systems in place do not work with AI; they might need to be changed or replaced fully to make way for parts of AI. Because AI is going to potentially change a lot in the way we think about risk, we have to be careful with it. We have to think about how powerful our computers are and should be, ease of explanation, good quality data, and resistance to adversarial attack.

5. Conclusion

AI has made finding, guessing, and lowering of security risks much easier when we test web apps for holes. AI-powered systems were much more accurate compared to older systems. They employ deep learning, machine learning, and reinforcement learning in finding things quicker and with fewer false positives. These traits make AI great for spotting threats both in real-time and before they happen. This is particularly true for banks and online stores, which are pretty hard to understand and have a lot of risk. AI also makes security enforcement and patch management easier by performing them automatically. This will let businesses move fast where there is danger while still following rules such as PCI-DSS and GDPR. After all these changes, AI-based solutions still continue to have problems. People cannot use deep learning models very often because they have so many problems. For example, AI requires a great amount of processing power, it is hard to figure out in making a choice, and attacks that are intended to hurt people happen more often.

We have to have systems that are stronger and more open because enemies find more ways to get around or trick the detection algorithms to take advantage of AI model flaws. Also, it is similarly hard for security experts to understand why a certain AI model did something or kept safe because many of them are incomprehensible. In cases where this counts, people may not be so responsible or trusting to others. Future research should hence be

focused on developing explainable AI frameworks which offer better clarity without compromising on accuracy. You can also develop a better defence plan by using hybrid models joining the best of AI, rule-based methods, and signature-based methods. By investing more money in better training datasets, making it harder for hackers to get into AI, and making current cybersecurity systems easier to use, AI is bound to become more useful and trustworthy in this area. As the rates of cyber threats are increasing along with difficulties in handling them, the importance of AI in keeping web applications safe is very likely to increase. The ability of finding weaknesses by AI systems may help us in finding better, more flexible, and stronger ways to keep hackers out of computers in the future. This would make it less safe to make and use web apps.

6. References

[1] Ahmad, M., Ghafoor, K. Z., Bakar, K. A., & Lloret, J. (2021). An AI-based vulnerability detection system for web applications. *IEEE Access*, *9*, 74082–74095. https://doi.org/10.1109/ACCESS.2021.3078513

[2] Shirazi, M., & Stojanovic, N. (2020). Machine learning approaches for web application vulnerability detection. *Journal of Information Security and Applications*, *55*, 102581. https://doi.org/10.1016/j.jisa.2020.102581

[3] Chakraborty, S., Alam, M., & Saha, S. (2022). Deep learning-based intrusion detection systems: A comprehensive review. *Computers & Security*, *113*, 102577. https://doi.org/10.1016/j.cose.2021.102577

[4] Sharma, P., Gupta, B., & Chatterjee, P. (2021). Use of AI in vulnerability management: Challenges and opportunities. *Future Generation Computer Systems*, *125*, 544–559. https://doi.org/10.1016/j.future.2021.07.002

[5] Kim, J., & Kim, H. (2020). Adversarial attacks on deep learning-based malware detection systems. *Security and Privacy*, *3*(1), e97. https://doi.org/10.1002/spy2.97

[6] Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017). Malware traffic classification using convolutional neural network for representation learning. *IEEE International Conference on Information Networking (ICOIN)*, 712–717. https://doi.org/10.1109/ICOIN.2017.7899588

[7] Caceres, J., Cuadrado-Gallego, J. J., & Gutiérrez, C. (2021). A hybrid machine learning model for the automatic detection of vulnerabilities in web applications. *Expert Systems with Applications*, *184*, 115500. https://doi.org/10.1016/j.eswa.2021.115500

[8] Liu, H., Lang, B., Liu, M., & Yan, H. (2020). CNN and RNN-based payload classification methods for web application security. *IEEE Transactions on Reliability*, *69*(3), 1124–1136. https://doi.org/10.1109/TR.2020.2968312

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

[10] Nasr, M., Shokri, R., & Houmansadr, A. (2018). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *IEEE Symposium on Security and Privacy*, 739–753. https://doi.org/10.1109/SP.2019.00065

[11] Zou, D., Wang, S., Han, Y., Jin, H., & Li, S. (2019). Reinforcement learning-based adaptive security configuration for cloud applications. *IEEE Transactions on Cloud Computing*, *9*(2), 545–558. https://doi.org/10.1109/TCC.2019.2908722

[12] Koloseni, D., Pham, H. V., & Kim, D. S. (2022). Explainable AI for security: A survey. *ACM Computing Surveys (CSUR)*, *55*(2), 1–41. https://doi.org/10.1145/3507907

[13] OWASP Foundation. (2021). *OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks*. https://owasp.org/Top10/

[14] Amankwa, E., Gyamfi, E., & Forkuo, E. K. (2023). Enhancing cybersecurity threat detection using AI and machine learning models. *Journal of Cybersecurity and Information Management*, *10*(1), 15–28.

[15] Bostani, H., & Sheikhan, M. (2017). Hybrid of anomaly-based and signature-based IDS for detecting unknown attacks. *Computer Networks*, *122*, 25–35. https://doi.org/10.1016/j.comnet.2017.05.021

[16] Gangineni, V. N., Pabbineedi, S., Penmetsa, M., Bhumireddy, J. R., Chalasani, R., & Tyagadurgam, M. S. V. (2022). Efficient Framework for Forecasting Auto Insurance Claims Utilizing Machine Learning Based Data-Driven Methodologies. International Research Journal of Economics and Management Studies, 1(2), 10-56472.

[17] Tyagadurgam, M. S. V., Gangineni, V. N., Pabbineedi, S., Penmetsa, M., Bhumireddy, J. R., & Chalasani, R. (2022). Designing an Intelligent Cybersecurity Intrusion Identify Framework Using Advanced Machine Learning Models in Cloud Computing. Universal Library of Engineering Technology, (Issue).

[18] Chalasani, R., Tyagadurgam, M. S. V., Gangineni, V. N., Pabbineedi, S., Penmetsa, M., & Bhumireddy, J. R. (2022). Leveraging Big Datasets for Machine Learning-Based Anomaly Detection in Cybersecurity Network Traffic. Available at SSRN 5538121.

[19] Bhumireddy, J. R., Chalasani, R., Tyagadurgam, M. S. V., Gangineni, V. N., Pabbineedi, S., & Penmetsa, M. (2022). Big Data-Driven Time Series Forecasting for Financial Market Prediction: Deep Learning Models. Journal of Artificial Intelligence and Big Data, 2(1), 153-164.

[20] Vangala, S. R., Polam, R. M., Kamarthapu, B., Kakani, A. B., Nandiraju, S. K. K., & Chundru, S. K. (2022). Leveraging Artificial Intelligence Algorithms for Risk Prediction in Life Insurance Service Industry. Available at SSRN 5459694.

[21] Chundru, S. K., Vangala, S. R., Polam, R. M., Kamarthapu, B., Kakani, A. B., & Nandiraju, S. K. K. (2022). Efficient Machine Learning Approaches for Intrusion Identification of DDoS Attacks in Cloud Networks. Available at SSRN 5515262.

[22] Polu, A. R., Narra, B., Buddula, D. V. K. R., Patchipulusu, H. H. S., Vattikonda, N., & Gupta, A. K. BLOCKCHAIN TECHNOLOGY AS A TOOL FOR CYBERSECURITY: STRENGTHS. WEAKNESSES, AND POTENTIAL APPLICATIONS.

[23] Nandiraju, S. K. K., Chundru, S. K., Vangala, S. R., Polam, R. M., Kamarthapu, B., & Kakani, A. B. (2022). Advance of AI-Based Predictive Models for Diagnosis of Alzheimer's Disease (AD) in healthcare. Journal of Artificial Intelligence and Big Data, 2(1), 141–152.DOI: 10.31586/jaibd.2022.1340

[24] Gangineni, V. N., Pabbineedi, S., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., & Tyagadurgam, M. S. V. (2023). AI-Enabled Big Data Analytics for Climate Change Prediction and Environmental Monitoring. International Journal of Emerging Trends in Computer Science and Information Technology, 4(3), 71-79.

[25] Pabbineedi, S., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., Tyagadurgam, M. S. V., & Gangineni, V. N. (2023). Scalable Deep Learning Algorithms with Big Data for Predictive Maintenance in Industrial IoT. International Journal of AI, BigData, Computational and Management Studies, 4(1), 88-97.

[26] Bhumireddy, J. R., Chalasani, R., Tyagadurgam, M. S. V., Gangineni, V. N., Pabbineedi, S., & Penmetsa, M. (2023). Predictive models for early detection of chronic diseases in elderly populations: A machine learning perspective. Int J Comput Artif Intell, 4(1), 71-79.

[27] Polam, R. M. (2023). Predictive Machine Learning Strategies and Clinical Diagnosis for Prognosis in Healthcare: Insights from MIMIC-III Dataset. Available at SSRN 5495028.

[28] Bhumireddy, J. R. (2023). A Hybrid Approach for Melanoma Classification using Ensemble Machine Learning Techniques with Deep Transfer Learning Article in Computer Methods and Programs in Biomedicine Update. Available at SSRN 5667650.

[29] Gupta, A. K., Polu, A. R., Narra, B., Buddula, D. V. K. R., Patchipulusu, H. H. S., & Vattikonda, N. (2024). Leveraging Deep Learning Models for Intrusion Detection Systems for Secure Networks. Journal of Computer Science and Technology Studies, 6(2), 199-208.

[30] Narra, B., Buddula, D. V. K. R., Patchipulusu, H., Vattikonda, N., Gupta, A., & Polu, A. R. (2024). The Integration of Artificial Intelligence in Software Development: Trends, Tools, and Future Prospects. Available at SSRN 5596472.

[31] Achuthananda, R. P., Bhumeka, N., Dheeraj Varun Kumar, R. B., Hari Hara, S. P., & Navya, V. (2024). Evaluating Machine Learning Approaches for Personalized Movie Recommendations: A Comprehensive Analysis. J Contemp Edu Theo Artific Intel: JCETAI-115.

[32] Polu, A. R., Narra, B., Buddula, D. V. K. R., Hara, H., Patchipulusu, S., Vattikonda, N., & Gupta, A. K. Analyzing the Role of Analytics in Insurance Risk Management: A Systematic Review of Process Improvement and Business Agility.

[33] Gangineni, V. N., Tyagadurgam, M. S. V., Pabbineedi, S., Penmetsa, M., Bhumireddy, J. R., & Chalasani, R. (2024). AI-Powered Cybersecurity Risk Scoring for Financial Institutions Using Machine Learning Techniques (Approved by ICITET 2024). Journal of Artificial Intelligence & Cloud Computing.

[34] Vangala, S. R., Polam, R. M., Kamarthapu, B., Kakani, A. B., Nandiraju, S. K. K., & Chundru, S. K. (2024). A Machine Learning-Based Framework for Predicting and Improving Student Outcomes Using Big Educational Data (Approved by ICITET 2024). Available at SSRN 5515379.