УЛОГА ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ У УНАПРЕЂЕЊУ САЈБЕР БЕЗБЕДНОСТИ

THE ROLE OF ARTIFICIAL INTELLIGENCE IN IMPROVING CYBERSECURITY

ЈЕЛЕНА МАРЈАНОВИЋ[1]
СТЕВАН СТАНКОВСКИ[2]

Оригинални научни рад
DOI: 10.5937/VI25103M

Резиме: Вештачка интелигенција (ВИ) постаје кључни фактор у унапређењу сајбер безбедности кроз све фазе животног циклуса софтвера – од развоја до продукције. У фази развоја софтвера, ВИ омогућава аутоматску анализу кода, детекцију рањивости и предиктивно моделирање ризика. Током тестирања и имплементације, користи се за симулацију напада и оптимизацију сигурносних процедура. У продукцији, ВИ обезбеђује континуирани мониторинг, детекцију аномалија и адаптивни одговор на претње у реалном времену. Овај рад истражује како ВИ унапређује сигурност софтверских система, трансформише традиционалне приступе заштити и поставља темеље за интелигентне системе сајбер одбране.

Кључне речи: Вештачка интелигенција, предикција, системи сајбер одбране

Abstract: Artificial Intelligence (AI) is becoming a key factor in advancing cybersecurity throughout all phases of the software lifecycle—from development to production. In the development phase, AI enables automated code analysis, vulnerability detection, and predictive risk modeling. During testing and implementation, it is used for attack simulation and optimization of security procedures. In production, AI provides continuous monitoring, anomaly detection, and adaptive real-time threat response. This paper explores how AI enhances the security of software systems, transforms traditional protection approaches, and lays the foundation for intelligent cyber defense systems.

Key Words: Artificial Intelligence, Prediction, Cyber defense systems

[1] Јелена Марјановић, Digital energy division, Schneider electric, Индустријска 3г, Нови Сад, jelena.marjanovic@se.com, ORCID: 0000-0002-8363-6228

[2] Стеван Становски, Универзитет у Новом Саду, Факултет техничких наука, Трг Доситеја Обрадовића 6, Нови Сад, stevan@uns.ac.rs, ORCID: 0000-0002-4311-1507

1. Увод

Развој софтверских система пролази кроз различите фазе животног циклуса, а избор одговарајуће методологије има кључну улогу у квалитету и ефикасности испоруке. Традиционални Waterfall модел представља линеаран и секвенцијалан приступ, где се свака фаза – од анализе захтева до одржавања – завршава пре преласка на следећу. Овакав приступ обезбеђује висок ниво документације и предвидљивости, али је мање флексибилан када дође до промена у захтевима пројекта [1]. У фази тестирања, вештачка интелигенција (ВИ) се користи за генерисање тест случајева и симулацију напада, док у продукцији обезбеђује континуирани мониторинг и детекцију аномалија. Међутим, ригидност Waterfall модела отежава брзу адаптацију нових ВИ алата, што представља ограничење у динамичним окружењима [6].

Насупрот томе, Agile методологија уводи итеративни и инкрементални развој, фокусиран на адаптивно планирање, сталну сарадњу са корисницима и брзу испоруку функционалних делова система. Agile омогућава тимовима да се брзо прилагођавају променама и смањују ризик неуспеха кроз континуиране повратне информације [2]. Agile методологија, заснована на итеративном развоју, пружа идеалан оквир за интеграцију ВИ. У раним спринтовима ВИ може аутоматски анализирати код и открити рањивости, док у каснијим итерацијама омогућава real-time threat intelligence и адаптивни одговор на претње. ВИ подржава континуирану процену ризика и оптимизацију сигурносних процедура, чиме се смањује ризик од пропуста у брзим циклусима испоруке [7], [8].

У последњој деценији, појављује се DevOps као еволуција Agile приступа, са циљем интеграције развоја и операција. DevOps наглашава аутоматизацију, континуирану интеграцију и испоруку (CI/CD), као и културу сарадње између тимова, чиме се убрзава процес испоруке и побољшава квалитет софтвера [3], [4]. Овај приступ не само да смањује време до тржишта, већ и омогућава брже откривање и отклањање грешака, уз повећање поузданости система [5]. DevOps, а посебно његов сигурносни проширени облик.

DevSecOps, представља најпогоднији оквир за примену ВИ. Интеграција ВИ у CI/CD pipeline омогућава:

- Аутоматско скенирање кода и детекцију рањивости пре продукције.
- Предикативну аналитику за идентификацију потенцијалних напада.
- Аутоматизовано тестирање и инцидент response, смањујући људске грешке и убрзавајући реакцију на претње. Овај приступ обезбеђује континуирану заштиту и адаптивни одговор у реалном времену, што је кључно за cloud-native и микросервисне архитектуре [9], [10].

У савременом окружењу, где се софтверски системи развијају у динамичним и комплексним условима, сајбер безбедност постаје кључни фактор квалитета и поузданости. Традиционални приступи заштити, засновани на статичким контролама и ручним проверама, више нису довољни да одговоре на софистициране претње. Вештачка интелигенција (ВИ) уводи парадигму проактивне и адаптивне заштите, интегрисане кроз све фазе животног циклуса софтвера (SDLC), узимајући у обзир различите методологије развоја.

2. Преглед доступних алата ВИ у безбедном развоју

Вештачка интелигенција постаје кључни елемент у унапређењу сајбер безбедности у свим фазама животног циклуса софтвера, а њен утицај је посебно видљив када се интегрише у различите методологије развоја – Waterfall, Agile и DevOps. Свака од ових методологија има специфичне карактеристике које обликују начин на који се безбедност имплементира, а ВИ алати омогућавају да се традиционални приступи трансформишу у проактивне, аутоматизоване и адаптивне процесе.

У Waterfall моделу, где се кораци извршавају секвенцијално, ВИ омогућава рано откривање безбедносних ризика и минимизирање трошкова касних измена. У Agile окружењу, које се заснива на честим итерацијама и брзој испоруци, ВИ подржава тимове кроз континуирану анализу кода, аутоматско тестирање и предвиђање ризика у сваком спринту. DevOps методологија, са својим нагласком на континуираној интеграцији, достави и надзору, додатно добија на вредности употребом ВИ у аутоматизацији безбедносних провера, анализи аномалија и откривању напада у реалном времену. Комбинација ових техника омогућава организацијама да граде безбедније, софтверске системе које је лакше одржавати и унапређивати у условима савременог развоја.

ВИ у Waterfall методологији

У фази пројектовања и развоја, ВИ алати попут Checkmarx и SonarQube користе машинско учење за статичку анализу кода и откривање рањивости пре него што софтвер пређе у следећу фазу [11]. Генеративни алати ВИ се све више користе за аутоматско генерисање сигурносних тестова и предвиђање потенцијалних ризика на основу историјских података и образаца у коду [12]. У тестирању, ВИ омогућава симулацију напада и динамичку анализу апликација (DAST), чиме се смањује ризик од касних открића рањивости. Google OSS-Fuzz је пример алата који користи ВИ за откривање грешака у open-source пројектима, што је посебно значајно у Waterfall окружењу где су касне промене скупе [13].

Поред тога, ВИ алати се све чешће интегришу у фазу прикупљања захтева, где помажу у идентификацији недоследности и потенцијалних функционалних празнина у документацији. У овој раној фази, системи засновани на обради природног језика (NLP) могу аутоматски анализирати спецификације и предложити побољшања или упозорити на двосмислене формулације које би касније могле довести до скупих измена. У фази имплементације, ВИ може аутоматски давати препоруке програмерима, нудити оптимизације кода у реалном времену и указивати на anti-patern који би могли угрозити архитектуру. На нивоу контроле квалитета, ВИ убрзава процес регресионог тестирања генерисањем оптимизованих скупова тестова који покривају критичне путање система. У оквиру финалне валидације, ВИ модели могу предвидети вероватноћу грешака након пуштања у рад и предложити додатне мере за побољшање стабилности система, што Waterfall процес чини предвидљивијим и ефикаснијим.

ВИ у Agile методологији

Агиле методологија, заснована на итеративном развоју и брзим циклусима, захтева континуирану интеграцију безбедносних провера. ВИ се овде користи за аутоматизацију тестирања, генерисање тест случајева и предиктивну анализу ризика у реалном времену. Алати попут Snyk и GitHub Advanced Security примењују машинско учење за идентификацију рањивости у зависностима током сваког спринта, док системи базирани на ВИ оптимизују приоритизацију backlog-а и планирање спринтова [4]. Студије показују да интеграција ВИ у Agile тестирање смањује време потребно за регресионе тестове и повећава покривеност тестирања, што је потврђено у случајевима компанија попут Google и Microsoft које користе ВИ за селекцију релевантних тестова у CI/CD окружењу [15].

Алати ВИ све више подржавају product owner-е и Scrum тимове у доношењу одлука, анализирајући историјске податке о перформансама тимова, брзини испоруке и учесталости дефеката. На основу ових анализа, ВИ може да предвиди оптерећење тима у наредним итерацијама и предложи оптималан број задатака за спринт.

У DevOps делу Agile екосистема, ВИ се користи за аутоматско препознавање аномалија у системским логовима, што омогућава брже откривање инцидената и смањује MTTR (Mean Time To Recovery). Такође, ВИ може препоручити који модули највероватније захтевају рефакторинг, на основу трендова деградације квалитета кода. Ова комбинација предикативне аналитике и аутоматизације чини Agile процесе значајно ефикаснијим, стабилнијим и отпорнијим на ризике који се јављају у динамичним развојним циклусима.

AI у DevOps методологији

DevOps приступ, који комбинује развој и операције, посебно погодује примени ВИ у фазама имплементације и продукције. ВИ се користи за континуирано праћење и детекцију аномалија у реалном времену кроз алате као што су Darktrace, CrowdStrike Falcon и Microsoft Defender for Endpoint, који примењују алгоритме машинског учења за откривање непознатих претњи и zero-day напада [16]. У DevSecOps окружењу, ВИ унапређује статичку (SAST) и динамичку анализу (DAST), као и интерактивно тестирање (IAST), омогућавајући аутоматску валидацију усклађености са стандардима попут GDPR, HIPAA и OWASP [17]. SOAR платформе, попут IBM Resilient и Cortex XSOAR, интегришу ВИ како би аутоматизовале одговор на инциденте, смањујући време реакције и повећавајући ефикасност тимова за безбедност [18]. Реални примери из индустрије показују да ВИ може смањити време имплементације за 30% и побољшати детекцију претњи за 45% у DevOps окружењу [19].

ВИ у DevOps окружењу омогућава и дубљу анализу телеметријских података из контејнерских и клауд окружења, где традиционални системи често не могу да идентификују суптилне индикаторе компромитовања. Модели машинског учења могу препознати необичне обрасце понашања микросервиса, нарушавање дозвољених токова саобраћаја или неправилности у аутентикацији, што омогућава брже изоловање инцидената. ВИ такође подржава аутоматско рангирање аларма, елиминишући велики број лажних позитивних резултата који традиционално оптерећују SOC тимове. Комбинација ових могућности омогућава организацијама да доследно одржавају висок ниво безбедности и стабилности током целокупног CI/CD процеса.

3. Предности и изазови

Примена вештачке интелигенције у безбедносним процесима доноси низ предности које значајно унапређују ефикасност и квалитет заштите софтверских система. Једна од кључних предности је бржа детекција претњи захваљујући алгоритмима машинског учења који анализирају велике количине података и идентификују аномалије у реалном времену [16], [20]. ВИ омогућава смањење лажних позитивних резултата кроз адаптивне моделе који уче из историјских инцидената и контекста, чиме се оптимизује рад тимова за безбедност [14]. Још једна важна предност је аутоматизација одговора на инциденте, коју омогућавају SOAR платформе интегрисане са ВИ, смањујући време реакције и повећавајући отпорност система [15]. Поред тога, ВИ доприноси предикативном планирању опоравка, анализирајући обрасце напада и превиђајући потенцијалне рањивости пре него што дође до компромитовања система [22].

Међутим, интеграција ВИ у ове методологије није без изазова. Недостатак квалитетних података за тренирање модела представља један од највећих проблема, јер неадекватни или пристрасни подаци могу довести до погрешних предикција и одлука [20]. Комплексност интеграције ВИ алата у постојеће процесе развоја и операција захтева додатне ресурсе, обуку и промену организационих структура [12], [14].

Такође, постоји ризик од adversarial ВИ напада, где нападачи манипулишу улазним подацима како би заварали моделе и избегли детекцију [20]. Ови изазови указују на потребу за објашњивом ВИ (Explainable AI), која омогућава транспарентност одлука и повећава поверење у аутоматизоване системе [21-22].

Будући трендови укључују развој генеративних ВИ алата за симулацију напада и креирање сигурносних playbook-а, као и интеграцију ВИ у квантно-отпорну криптографију, што ће додатно унапредити отпорност система на софистициране претње [22]. Очекује се да ће комбинација ових технологија омогућити проактивну сајбер одбрану засновану на анализи великих скупова података и адаптивним механизмима заштите [11], [16].

4. Закључак

Интеграција вештачке интелигенције у методологије развоја софтвера – Waterfall, Agile и DevOps – представља кључну трансформацију у области сајбер безбедности. ВИ не само да унапређује традиционалне приступе, већ уводи проактивне и адаптивне механизме заштите који омогућавају бржу детекцију претњи, аутоматизацију процеса и предикативну аналитику.

У Waterfall окружењу, ВИ доприноси смањењу ризика кроз статичку анализу кода и симулацију напада, док у Agile методологији обезбеђује континуирану верификацију безбедности у итеративним циклусима. DevOps приступ, са нагласком на аутоматизацију и оркестрацију, посебно профитира од примене ВИ кроз real-time мониторинг, детекцију аномалија и аутоматизован одговор на инциденте.

Предности овакве интеграције су јасне: повећана ефикасност, смањење лажних позитивних резултата и већа отпорност система на софистициране нападе. Ипак, изазови попут недостатка квалитетних података, комплексности имплементације и ризика од напада на ВИ моделе захтевају даља истраживања и развој објашњивих и поузданих ВИ решења.

Будућност сајбер безбедности лежи у комбинацији ВИ технологија са напредним криптографским методама и предикативним механизмима одбране, што ће омогућити стварање интелигентних, адаптивних и отпорнијих система.

5. Изјаве захвалности

Истраживања у овом раду су финансирана са пројекта под називом „Унапређење квалитета наставе на судијским програмима Департмана кроз имплементацију резултата научно-истраживачког рада у области Индустријског инжењерства и менаџмента“.

У припреми овог рада су коришћени ChatGPT, Gemini и Copilot.

6. Литература

[1] D. B. J. Rao, P. Kalpana, G. S. N. Kumar, and N. M. Atcha, A Comparative Analysis of Software Development Models: Waterfall, Agile and DevOps, *Intelligent Systems and Sustainable Computing*, Springer, pp. 589–599, 2025.

[2] V. Radovanović et al., Comparison of Agile and DevOps Methodologies: Analysis of Efficiency, Flexibility, and Application in Software Development, *JITA*, vol. 14, pp. 78–83, 2024.

[3] R. Amaro, R. Pereira, and M. M. da Silva, Mapping DevOps capabilities to the software life cycle: A systematic literature review, *Information and Software Technology*, vol. 177, p. 107583, 2025.

[4] M. Gokarna and R. Singh, DevOps: A Historical Review and Future Works, *arXiv preprint*, 2020.

[5] M. E. Misri, The Impact Of DevOps Practices On Software Development Lifecycle, *IJRTI*, 2024.

[6] H. U. Khan et al, AI-driven cybersecurity framework for software development based on the ANN-ISM paradigm, *Scientific Reports*, vol. 15, pp. 1–18, 2025.

[7] A. H. Salem et al, Advancing cybersecurity: a comprehensive review of AI-driven detection techniques, *Journal of Big Data*, vol. 11, Article 105, 2024.

[8] N. Mohamed, Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms, *Knowledge and Information Systems*, vol. 67, pp. 6969–7055, 2025.

[9] I. Coston et al, Enhancing Secure Software Development with AZTRM-D: An AI-Integrated Approach Combining DevSecOps, Risk Management, and Zero Trust, *Applied Sciences*, vol. 15, no. 15, p. 8163, 2025.

[10] F. Binbeshr and M. Imam, Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions, *arXiv preprint*, Apr. 2025.

[11] F. Binbeshr and M. Imam, Comparative Analysis of AI-Driven Security Approaches in DevSecOps, *arXiv preprint*, 2025.

[12] M. Prakash, *Role of Generative AI Tools in SDLC – Waterfall Model*, MIT Thesis, 2024.

[13] Google, OSS-Fuzz Project, Available: https://google.github.io/oss-fuzz/, ,accessed 20.11.2025.

[14] A. C. Figueiredo, R. Pereira, and M. A. Silva, *Exploring the Integration of AI and DevOps for Agile Product Development*, Springer, 2025.

[15] DigitalDefynd, AI in Software Testing: 5 Case Studies, https://digitaldefynd.com/IQ/ai-in-software-testing-case-studies/, accessed 20.11.2025.

[16] E. Lee, How AI Is Transforming Cybersecurity: Real-World Examples, *TechBullion*, 2025.

[17] T. Abdiukov, Automated Security Testing in DevSecOps Pipelines, *WJARR*, 2024.

[18] IBM, Resilient SOAR Platform, https://www.ibm.com/security/resilient, accessed 20.11.2025.

[19] S. Amgothu and G. Kankanala, AI/ML – DevOps Automation, *AJER*, 2024.

[20] C. Dilmegani, Top 13 AI Cybersecurity Use Cases with Real Examples, *AIMultiple*, 2025.

[21] Stankovski S, EIM'S MESSAGE: Be Ready to Incorporate AI Principles into Education, *Journal of Mechatronics Automation and Identification Technology,* vol. 10, no. 2. pp. E1-E2, 2025.

[22] R. Garvey, The Impact of AI on Waterfall and Agile Methodologies, https://www.linkedin.com/pulse/impact-ai-waterfall-agile-methodologies-pros-cons-ross-garvey-0nlxe/, accessed 20.11.2025.