DOI: https://doi.org/10.15276/ict.01.2024.31
УДК 004.4’24

Інтеграція обробки природної мови та методів машинного навчання для забезпечення безпеки смарт-контрактів: порівняння з класичними методами

Терещенко Олександр Ігорович[1)]
Аспірант каф. Інженерії програмного забезпечення
ORCID: https://orcid.org/0000-0003-4510-5255; alexandr.tereschenko2014@gmail.com.
Scopus Author ID: 57705566400
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

У сучасних блокчейн-системах смарт-контракти є одним із найважливіших компонентів для забезпечення автоматизованого виконання угод без необхідності посередників. Проте, смарт-контракти, написані на таких мовах, як Solidity, можуть містити вразливості, які можуть бути використані зловмисниками для викрадення коштів або маніпуляцій з активами. Зважаючи на зростаючу кількість атак на смарт-контракти, розробка ефективних методів для їх виявлення є критично важливою. Традиційні підходи до виявлення вразливостей у смарт-контрактах включають символьне виконання, фаззинг, формальну верифікацію та пошук шаблонів. Ці методи мають свої переваги, проте стикаються з низкою проблем, таких як високе споживання ресурсів, обмеженість у виявленні нових типів вразливостей та неможливість масштабування на великі контракти. У зв'язку з цим виникає необхідність у впровадженні нових підходів, таких як обробка природної мови (NLP) та машинне навчання, які можуть ефективніше вирішувати ці завдання. У даному дослідженні було розглянуто метод на основі NLP, який використовує Word2Vec для перетворення коду смарт-контракту у векторні представлення, що дозволяє краще аналізувати семантичні зв’язки між елементами коду. Далі ці векторні представлення подаються на вхід двонаправленої рекурентної нейронної мережі з блоками GRU та механізмом уваги. Такий підхід дозволяє моделі фокусуватися на найбільш важливих частинах коду та покращувати точність виявлення вразливостей. Проведений порівняльний аналіз показав, що методи на основі NLP значно перевершують класичні підходи в усіх ключових метриках. Зокрема, модель GRU з механізмом уваги показала високі результати в точності, повноті та F-міри, що робить її ефективною для виявлення складних вразливостей, таких як повторний вхід. Крім того, підхід на основі NLP має здатність адаптуватися до нових типів атак завдяки навчанню на великих наборах даних. Таким чином, інтеграція NLP та машинного навчання є перспективним напрямом для підвищення безпеки смарт-контрактів. Подальші дослідження можуть бути спрямовані на вдосконалення цих підходів, зокрема через впровадження новітніх моделей, таких як трансформери.

Ключові слова: смарт-контракт; блокчейн; глибоке навчання; виявлення вразливостей; NLP; машинне навчання; символьне виконання; фаззинг, формальна верифікація; пошук шаблонів

Актуальність. З розвитком технології блокчейн смарт-контракти стали невід'ємною частиною багатьох фінансових та інших транзакційних систем. Смарт-контракти керують значними ресурсами, що робить їх привабливими для зловмисників. Однією з найпоширеніших вразливостей є атака повторного входу, яка може призвести до значних економічних втрат. Класичні методи, такі як символьне виконання, фаззинг, формальна верифікація та пошук шаблонів, є основними підходами для виявлення вразливостей, але вони мають свої обмеження, особливо щодо адаптивності до нових типів атак. Тому виникає потреба у нових підходах, які можуть підвищити ефективність виявлення вразливостей. У цьому контексті інтеграція NLP з методами машинного навчання є перспективним напрямом досліджень.

Метою дослідження є порівняння ефективності класичних методів виявлення вразливостей у смарт-контрактах з методами, що базуються на обробці природної мови та машинному навчанні. В основі дослідження лежить аналіз продуктивності моделей на базі Word2Vec і GRU з механізмом уваги, а також їх порівняння з традиційними підходами.

Смарт-контракти піддаються широкому спектру атак через наявність вразливостей у їхньому коді. Одна з найбільш поширених вразливостей – це вразливість повторного входу (reentrancy). Вона виникає, коли зловмисник може повторно викликати функцію смарт-контракту до того, як попередній виклик завершить своє виконання.

Це дозволяє зловмисникам маніпулювати балансом контракту та, наприклад, вивести більше коштів, ніж було передбачено. Атака на “The DAO” у 2016 році є одним із найвідоміших прикладів використання цієї вразливості, що призвело до втрат майже 60 мільйонів доларів [1].

Ще однією розповсюдженою вразливістю є вразливість переповнення та зменшення числових значень (integer overflow/underflow). Ця вразливість виникає, коли арифметичні операції в контракті призводять до результатів, які виходять за межі допустимого діапазону числових типів даних. Це може дозволити зловмиснику, використовуючи переповнення або зменшення чисел, змінювати значення перемінних таким чином, що це порушує логіку контракту [2].

Небезпека також може виникнути в результаті використання невірного порядку операцій у смарт-контракті. Наприклад, якщо зміни в стані контракту відбуваються після зовнішніх викликів, це може відкрити можливості для експлуатації контракту, оскільки інші контракти можуть взаємодіяти зі зміненим станом, перш ніж будуть застосовані всі обмеження. Така проблема відома як вразливість "call to untrusted contracts", яка дозволяє зловмисникам скористатися зворотними викликами для перехоплення керування [3].

Існують також вразливості, пов'язані з неправильним використанням випадкових чисел. Оскільки на блокчейні важко створити дійсно випадкові числа, деякі контракти використовують передбачувані алгоритми для їх генерації, що може бути експлуатоване зловмисниками. Наприклад, зловмисник може передбачити майбутні результати «випадкових» значень і використати це для отримання несправедливих переваг.

Додатково, проблема зловживання дозволами (access control) є ще одним критичним аспектом, де контракти можуть мати помилки у механізмах обмеження доступу, що дозволяє зловмисникам отримувати несанкціонований доступ до важливих функцій, таких як зміна власника контракту або передача коштів. Це може стати причиною серйозних порушень безпеки, особливо у контрактах, що керують великими сумами активів [4].

Вразливості смарт-контрактів продовжують розвиватися, а зловмисники постійно шукають нові шляхи для експлуатації недоліків у коді, що робить забезпечення їхньої безпеки критично важливим завданням для розробників та дослідників.

Для порівняння ефективності класичних методів виявлення вразливостей у смарт-контрактах з методами на основі обробки природної мови (NLP) та машинного навчання було проведено серію експериментів на основі реальних даних, а саме вихідних кодів смарт-контрактів на мові Solidity з платформи Ethereum. Було використано кілька підходів, які різняться як за методами аналізу, так і за технологічними підходами.

Інструмент Mythril використовує метод символьного виконання смарт-контрактів з метою виявлення всіх можливих шляхів виконання, що можуть призвести до вразливостей. Подібним чином працює і Manticore [5]. Метод символьного виконання дозволяє знайти складні логічні помилки, які можуть виникнути лише за певних умов. Однак, цей метод часто стикається з проблемою вибуху станів, що значно збільшує обчислювальні витрати та час аналізу [6].

Для виявлення вразливостей методом фазингу було використано інструмент Echidna, який подає випадкові або спеціально згенеровані вхідні дані на програму. Цей підхід ефективно виявляє помилки, здатні призвести до відмови або експлуатації вразливостей, проте не завжди охоплює всі можливі шляхи виконання, що може призводити до пропуску певних вразливостей.

Формальна верифікація виконувалася за допомогою інструменту VeriSolid, який використовує математичні методи для підтвердження відповідності смарт-контракту заданим специфікаціям або інваріантам. Цей метод забезпечує високий рівень надійності, але потребує значних ресурсів для побудови і перевірки формальних моделей, що робить його менш практичним для великих або складних контрактів [7].

Інструмент Slither застосовує метод пошуку шаблонів, який використовує заздалегідь визначені шаблони для пошуку відомих вразливостей у коді смарт-контрактів. Цей підхід є

швидким і простим у застосуванні, але його обмеження полягають у залежності від наявних шаблонів, через що він може пропускати нові або модифіковані типи вразливостей [8].

Зазначені інструменти були використані для проведення порівняльного аналізу, який дозволив визначити, які типи вразливостей вони виявляють. Було встановлено, що всі інструменти підтримують виявлення лише двох типів вразливостей – повторного входу та переповнення/зменшення числових значень. У зв'язку з цим у дослідженні було зосереджено увагу саме на цих вразливостях. Результати аналізу наведені в Таблиці, де детально порівнюються можливості кожного інструменту щодо виявлення різних типів вразливостей.

NLP та машинне навчання, у свою чергу, пропонують новий підхід до аналізу смарт-контрактів. У цьому підході код смарт-контрактів спочатку перетворюється на векторні представлення за допомогою моделей, таких як Word2Vec. Далі ці векторні представлення подаються на вхід нейронних мереж. Цей підхід дозволяє моделі "розуміти" семантичні зв’язки між різними частинами коду, що підвищує ефективність виявлення складних вразливостей, таких як повторний вхід (reentrancy). Однією з основних переваг цього підходу є його здатність адаптуватися до нових типів атак, завдяки навчанню на великих наборах даних [6].

Для підвищення ефективності моделі, у якості входу нейронних мереж було використано двонаправлену рекурентну нейронну мережу з блоками GRU, що дозволяє використовувати інформацію як з попередніх, так і з наступних станів послідовності. Це допомагає моделі враховувати як початковий контекст, так і подальший розвиток подій у смарт-контракті, що є важливим для правильного визначення вразливостей. GRU-блоки використовуються замість LSTM через меншу кількість параметрів, які необхідно оптимізувати, що зменшує час тренування моделі. На рис. 1 наведена загальна архітектура запропонованої моделі.

Таблиця. **Порівняльний аналіз інструментів**

Інструмент	**Повторний вхід**	**Переповнення/зменшення числових значень**	**Невірний порядок операцій**	**Невірне використання випадкових чисел**	**Проблеми з доступом**
Mythril	+	+	+	-	+
Manticore	+	+	+	-	+
Echidna	+	+	-	+	-
VeriSolid	+	+	+	-	+
Slither	+	+	+	-	+

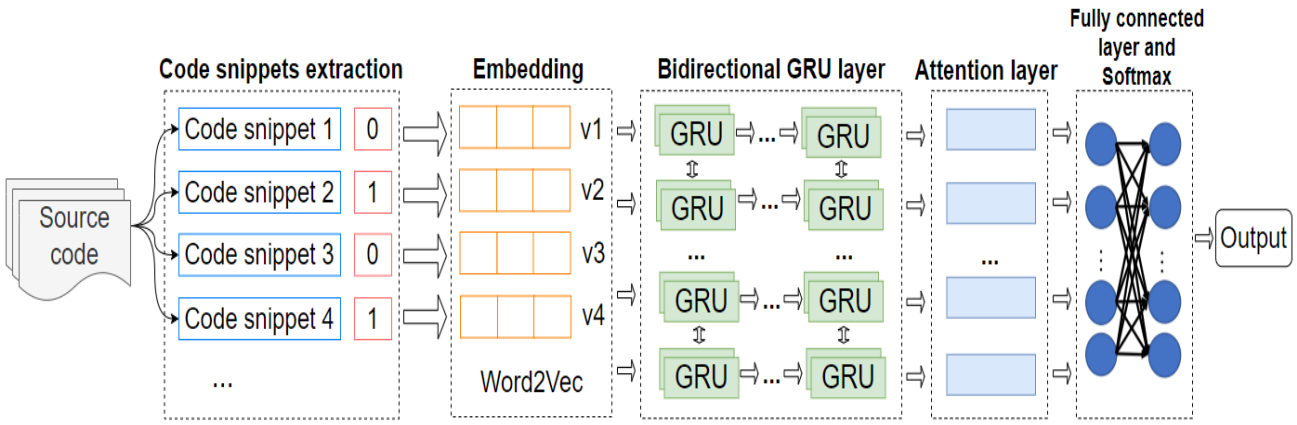

Рис. 1. **Архітектура Bidirectional GRU з механізмом уваги**

Крім того, був доданий механізм уваги, який фокусується на важливих частинах коду смарт-контракту, що мають найбільший вплив на виявлення вразливостей. Це дозволяє моделі концентрувати свою увагу на тих ділянках, де найчастіше виникають вразливості, таких як функції, пов'язані з передачею коштів між контрактами. Механізм уваги також покращує розуміння семантики коду, що зменшує кількість помилкових позитивних результатів.

Архітектура запропонованої моделі починається з перетворення коду смарт-контрактів у векторні представлення, що здійснюється за допомогою методу Word2Vec. Це дозволяє коду бути інтерпретованим у вигляді числових векторів, які можуть бути подані на вхід нейронних мереж. Наступним кроком є передача цих векторних представлень на вхід двонаправленої GRU мережі, яка здатна обробляти послідовні дані в обох напрямках, що забезпечує краще розуміння контексту смарт-контракту.

Далі у моделі використовується механізм уваги, який дозволяє зосередитися на найбільш важливих частинах коду, що підвищує точність виявлення вразливих фрагментів. Завдяки цьому модель може краще визначити ті ділянки коду, де найімовірніше знаходяться вразливості, такі як повторний вхід (reentrancy). На фінальному етапі відбувається класифікація смарт-контракту за допомогою лінійного шару з функцією Softmax, яка визначає, чи є контракт вразливим, чи безпечним. Такий підхід показав значне підвищення точності та адаптивності у порівнянні з іншими моделями, зокрема простими рекурентними мережами та моделями LSTM, за рахунок меншої кількості параметрів для оптимізації та ефективнішої роботи з послідовностями даних [9].

Порівняльний аналіз показав, що метод символьного виконання показав високу здатність до виявлення логічних помилок, але зіткнувся з проблемами масштабованості, особливо на великих контрактах, де аналіз усіх можливих шляхів виконання виявився надто дорогим з точки зору ресурсів. Метод фаззингу був ефективним для виявлення загальних помилок, але втратив ефективність у випадках, коли вразливість виникала лише за специфічних умов або комбінацій вхідних даних. Метод формальної верифікації забезпечив високу точність, але потребував значних ресурсів для побудови формальних моделей, що робить цей метод менш придатним для широкого використання. Метод пошуку шаблонів забезпечив швидке виявлення відомих вразливостей, але втратив ефективність у випадках нових або модифікованих атак.

NLP та машинне навчання показали найвищі результати в усіх категоріях, зокрема, у виявленні нових типів вразливостей завдяки використанню моделі Word2Vec та двонаправлених GRU з механізмом уваги. За даними [2], цей підхід показав F-міру на рівні 91.41 %, що перевищує результати інших методів. NLP-техніки мають кілька ключових переваг над класичними методами: адаптивність до нових вразливостей завдяки машинному навчанню на великих наборах даних, здатність глибше аналізувати семантичні зв’язки в коді, що підвищує точність виявлення складних вразливостей, та зменшення кількості помилкових позитивів.

Інтеграція NLP-технік з методами машинного навчання відкриває нові можливості для підвищення безпеки смарт-контрактів. У порівнянні з класичними методами, такими як символьне виконання, фаззинг, формальна верифікація та пошук шаблонів, підходи на основі NLP показують кращі результати, що дозволяє знижувати ризики, пов'язані з вразливостями у смарт-контрактах. Подальші дослідження можуть бути спрямовані на інтеграцію інших NLP-технік, таких як трансформери, для ще більшої оптимізації процесу виявлення вразливостей.

СПИСОК ЛІТЕРАТУРИ

1. Komleva N. O., Tereshchenko O. I. “Requirements for the development of smart contracts and an overview of smart contract vulnerabilities at the Solidity code level on the Ethereum platform.” *Herald of Advanced Information Technology*. 2023; 6 (1): 54–68. DOI: https://doi.org/0.15276/hait.06.2023.4.

2. Tereshchenko O., Komleva N. “Vulnerability Detection of smart contracts based on bidirectional GRU and attention mechanism”. *ICTERI 2023. Communications in Computer and Information Science. Cham*. 2023; 1980: 276–287.

3. Feng Q. et al. “Neural network-based graph embedding for cross-platform binary similarity detection”. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, Dallas*. 2017. p. 363.

4. Harer J. et al. “Learning to repair software vulnerabilities with generative adversarial networks”. *Advances in Neural Information Processing Systems. NIPS, San Francisco*. 2018. p. 7933.

5. Soydaner D. “Attention mechanism in neural networks: where it comes and where it goes”. *Neural Computing and Applications*. 2022; 34: 13371–13385. DOI: https://doi.org/10.1007/s00521-022-07366-3.

6. Sherstinsky A. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network”. *Physica D: Nonlinear Phenomena*. 2020; 404. DOI: https://doi.org/10.1016/j.physd.2019.132306.

7. Wisam A., Musa M., Bilal I. “An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges”. *International Engineering Conference (IEC)*. 2019. p. 200−204. DOI: https://doi.org/10.1109/IEC47844.2019.8950616.

8. Ferreira J., Cruz P., Durieux T., Abreu R. “SmartBugs: A Framework to Analyze Solidity Smart Contracts”. *35th IEEE/ACM International Conference on Automated Software Engineering (ASE 2020)*. Melbourne. 2020. DOI: https://doi.org/10.48550/arXiv.2007.04771.

9. Akca S., Rajan A. & Peng C. “SolAnalyser: A framework for analysing and testing smart contracts”. *26th Asia-Pacific Software Engineering Conference (APSEC)*. 2019. p. 482–489. DOI: https://doi.org/10.1109/APSEC48747.2019.00071.

DOI: https://doi.org/10.15276/ict.01.2024.31
UDC 004.4’24

Integration of NLP and machine learning methods for smart contract security: a comparison with traditional approaches

Oleksandr I. Tereshchenko[1)]
Postgraduate student, Department of Software Engineering
ORCID: https://orcid.org/0000-0003-4510-5255; alexandr.tereschenko2014@gmail.com
Scopus Author ID: 57705566400
[1)] Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

In modern blockchain systems, smart contracts are one of the most critical components for ensuring the automated execution of agreements without the need for intermediaries. However, smart contracts written in languages like Solidity may contain vulnerabilities that can be exploited by malicious actors to steal funds or manipulate assets. Given the increasing number of attacks on smart contracts, the development of effective methods for detecting such vulnerabilities is crucial. Traditional approaches to detecting vulnerabilities in smart contracts include symbolic execution, fuzzing, formal verification, and pattern matching. These methods have their advantages but face several challenges, such as high resource consumption, limitations in detecting new types of vulnerabilities, and difficulties in scaling to large contracts. As a result, there is a need to introduce new approaches, such as natural language processing (NLP) and machine learning, which can address these challenges more effectively. In this study, an NLP-based method was explored, using Word2Vec to convert smart contract code into vector representations, allowing for better analysis of the semantic relationships between elements of the code. These vector representations are then fed into a bidirectional recurrent neural network with GRU blocks and an attention mechanism. This approach allows the model to focus on the most important parts of the code and improve the accuracy of vulnerability detection. The comparative analysis showed that NLP-based methods significantly outperform traditional approaches in all key metrics. In particular, the GRU model with an attention mechanism demonstrated high results in accuracy, recall, and F-measure, making it effective for detecting complex vulnerabilities such as reentrancy. Furthermore, the NLP-based approach is capable of adapting to new types of attacks thanks to training on large datasets. Thus, the integration of NLP and machine learning represents a promising direction for enhancing the security of smart contracts. Future research can focus on improving these approaches, particularly through the implementation of advanced models such as transformers.

Keywords: Smart contract; Blockchain; deep learning; vulnerability detection; NLP; machine learning; symbolic execution; fuzzing; formal verification; pattern matching