




# DOORchain: Deep Ontology-Based Operation Research to Detect Malicious Smart Contracts

Mohamed A. El-Dosuky<sup>1</sup>(✉)  and Gamal H. Eladi<sup>2</sup>

<sup>1</sup> Computer Science Department, Mansoura University,  
Mansoura P.O. 35516, Egypt  
Mouh\_sal\_010@mans.edu.eg

<sup>2</sup> Information Systems Department, Mansoura University,  
Mansoura P.O. 35516, Egypt  
gamalhelmy@mans.edu.eg

**Abstract.** Blockchains have become of great vogue in different fields after the introduction of Bitcoin. There are some inherent problems that need to be solved. One of these problems is to ensure that secured transactions in blockchain are checked if they are malicious or not. This paper proposes DOORchain that combines three powerful approaches of detecting intrusions and maliciousness. They are Deep learning, Ontology, and Operation Research. This uses the advantage of constraints from operation research to formalize and detect network maliciousness, and ontology to detect behavioral maliciousness in particular. Then it feeds this formalization to deep learning in order to check if the transactions in blockchain are malicious or not. After applying the proposed DOORChain, the final results affirm that accuracy and recall are enhanced with a slight inescapable trade-off in precision.

**Keywords:** Smart contracts · Deep learning · Ontology · Operation Research · Blockchain

## 1 Introduction

Blockchain, a distributed ledger, could be contemplated as a data structure that allows for creating signing, sharing, and storing digital transactions in a manner that makes it tremendously difficult to alter or delete blocks once recorded on ledger [1]. A smart contract is capable of both defining agreement obligations, and enforcing those [2].

The *problem* is that blockchain users may make malicious transactions and we don't know if it trustworthy or not. This paper proposes a solution to this problem by developing a methodology that combines three powerful approaches of detecting intrusions and maliciousness. This uses the advantage of constraints from Operation Research (OR) to formalize and detect network maliciousness, ontology to detect behavioral maliciousness, and deep learning, as overviewed in Sect. 2, before presenting, evaluating, and concluding proposed methodology in subsequent sections.

## 2 Previous Work

Vulnerabilities of smart contracts are investigated [3]. A recent list of attacks and their counter-measures is devised [4]. An audacious attempt to model contracts is proposed [5]. Decentralization scanning of anti-malware is possible [6].

Among the powerful approaches of detecting intrusions and maliciousness there are constraints from OR, ontology-based approaches to detect behavioral maliciousness, and deep learning, as overviewed in the following subsections.

### 2.1 Operation Research

Constraints, from OR: a Swiss-knife of solving problems [7], can be used in detecting network intrusions [8] as shown in both NeMODE [9] for DSL [10], and PRIDE [11] for wireless networks. This is comparable and yet superior to packet classification [12] built on Snort [13].

### 2.2 Ontology

Ontology provides semantics-awareness to applications. Recently, it is proposed for malware behavioral analysis and attack detection [14]. For a recent detailed survey on its application in security, refer to [15].

### 2.3 Deep Learning

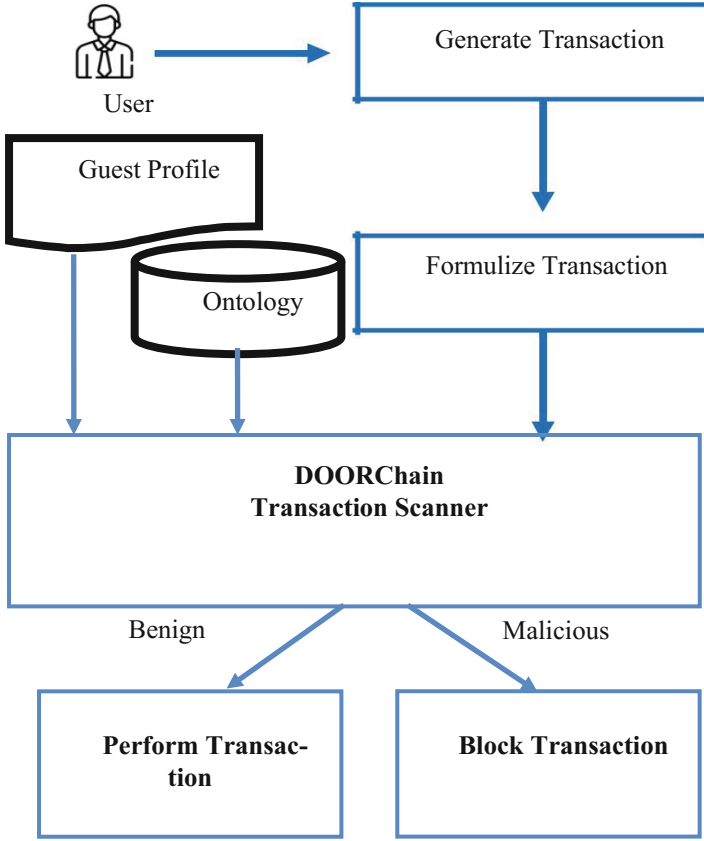
*Deep learning*, the state-of-the-art in multimedia processing, comprehend data with multiple levels of abstraction [16]. Regarding paper scope, there are DeepChain [17], and a clever inspection algorithm that is inspired by colors [18].

## 3 Proposed Methodology

The proposed methodology can make a transaction and test if it is trustworthy or malicious based on Deep learning, Ontology, and OR, as shown in Fig. 1.

If the transaction is benign (trustworthy), the network will pass it and it will be completed. If the transaction is malicious, the network will stop it because of its maliciousness. DOORchain acts as a 3-layer filter. Decision is based on OR first, then Ontology-based, and Deep learning at last.

The order of Deep [Ontology[OR[chain]]] is logical since the transaction is generated first (as shown in Subsect. 3.1), then the network packets are filtrated using OR (as in Subsect. 3.2), then behavioral maliciousness is detected by Ontology-based approaches (as shown in Subsect. 3.3), and finally the bytecode of smart contract per se is transformed into an encoded image and classified into either benign or malicious based on Deep learning (as shown in Subsect. 3.4).



**Fig. 1.** Block diagram of proposed methodology.

### 3.1 Generating Transaction

Beside the standard Javascript and Solidity [19], the authors have a tendency to use Python wherever possible. There are a plethora of Python packages to operate on Ethereum, such as web3.py [20] and Pyethereum [21]. Basic log analysis is implemented based on OYENTE [2]. The goal is to detect abuse attempts based on outlier patterns in gas consumption and transaction longevity.

### 3.2 Operation Research

The mathematical formalization is based on nomenclature shown in Table 1.

Equation (1) formalizes transaction flow to be maximized.

$$\Phi = \mu B + (1 - \mu)M \quad (1)$$

$\mu$  is a model parameter taking a value in the range [0..1]. Let us contemplate on its value at the two extremes of the business-as-usual scenario (at  $\mu = 1$ , the case of full

**Table 1.** Nomenclature

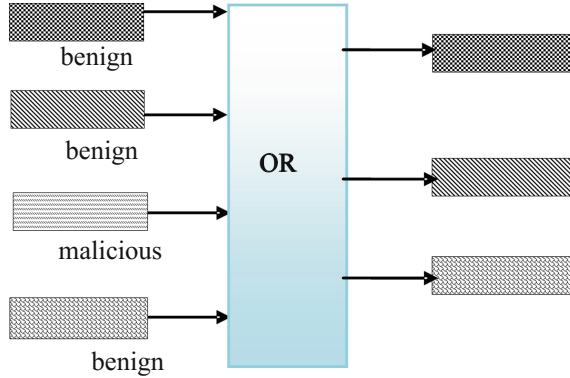
Symbol	Quantity
$\Phi$	Transaction flow
B	Benign (Trust) mode
M	Maliciousness mode
$\mu$	Model parameter
$T_i$	Transaction with id i
$N$	Total number of transactions

trust, with no maliciousness detected) and the paranoia scenario (at  $\mu = 0$ , the case of no trust, with high rate of maliciousness detection, or susceptibility of suspicion).

To filter out intrusion signatures, network packets are scanned in the network traffic logs, either generated from previous step (Sect. 3.1) or those downloaded from tcp-dump [22]. Each Transaction is generated with benign default mode (B). Equation (2) formalizes maliciousness mode.

$$\forall i, 1 \leq i \leq N \quad (T_i \cap \text{MODEL} = ? \varphi) \rightarrow \text{MODE}[T_i] := M \quad (2)$$

Where  $\varphi$  is empty set, and MODEL is an OR model built in NeMODE front-end to recognize the attack of SYN flood [23] and two Man-in-The-Middle attacks [24], namely DHCP and DNS spoofing. Figure 2 depicts filtrating network packets.

**Fig. 2.** Modeling the filtration of network packets.

### 3.3 Ontology

The ontology is built with Protégé [25] after [14], taking into account common backdoor, Trojans, and worms, that infect files, registries, or networks as shown in Fig. 3. A key incorporated concept in the ontology is the elusive Ransomware [26].

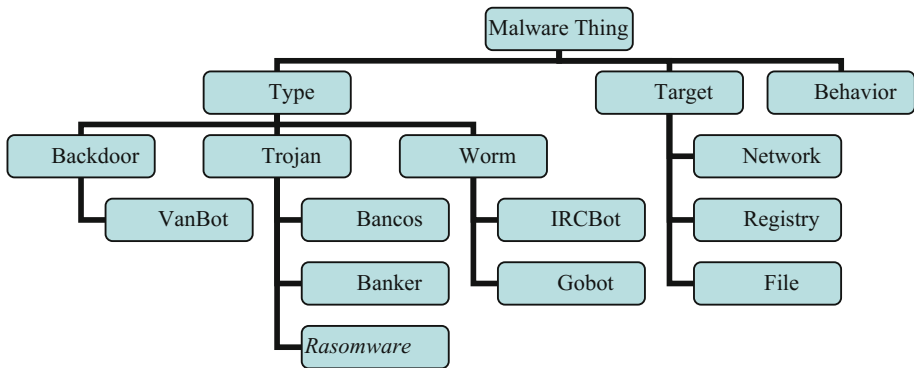


Fig. 3. Malware ontology

3.4 Deep Learning

This component follows the steps from [18]. However, the architecture of the built and deployed convolutional neural network (CNN) to the backend is adopted from [27]. The flow chart is depicted in Fig. 4.

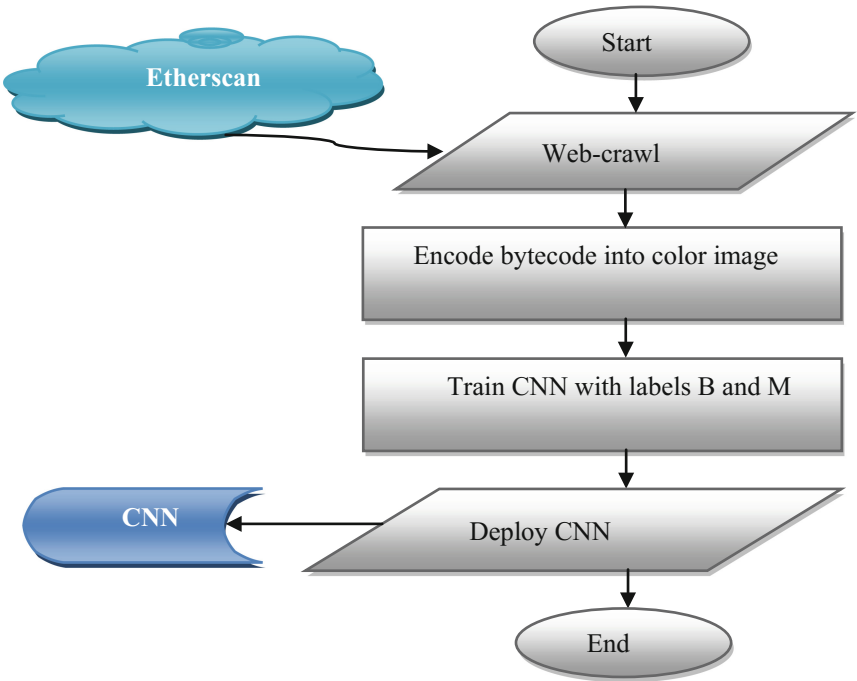


Fig. 4. Flow chart of deep learning component; based on [18]

4 Experiments and Results

Unless otherwise stated, development of proposed methodology took place on either of relatively similar machines whose specifications are shown in Table 2.

Table 2. Hardware specs

Feature	MachineA	MachineB
Manufacturer	Dell	Acer
Model	Inspiron N5040	Extensa 5630
Processor	Intel Pentium 2.1 GHz	Intel Pentium 2.2 GHz
Memory	2048 Mb	3027 Mb
Operating system	Windows 7 32bits	Windows 10 32bits

Smart contract web-crawling is achievable using pyspider [28], then the bytecode can be encoded as in Fig. 5.

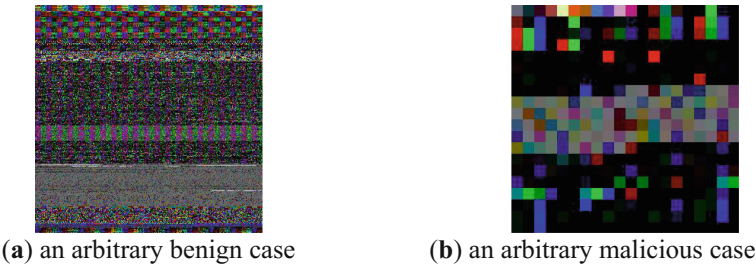


Fig. 5. Arbitrary encoded images of smart contract bytecode. Dataset from [29].

To compare DOORchain, CNN is built on the same configuration in [18] on the same dataset [29]. The comparison is depicted in Fig. 6.

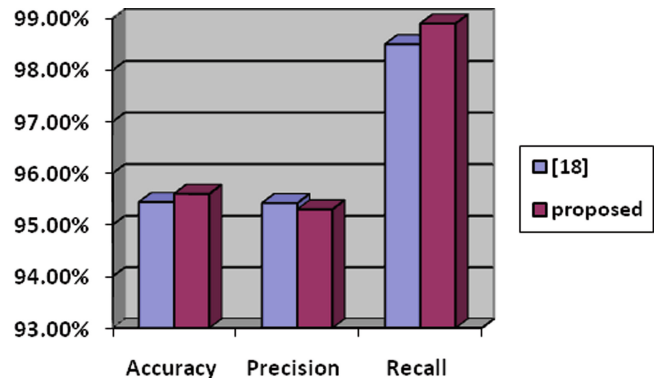


Fig. 6. Comparison between DOORChain and a previous work

After applying the proposed DOORChain, the final results affirm that accuracy and recall are enhanced with a slight inescapable trade-off in precision.

## 5 Conclusion and Future Directions

The proposed methodology tries to scan smart contracts and filtrate out malicious transactions. It acts as a 3-layer filter based on Deep learning, Ontology, and OR.

One possible future direction could be to combine decentralized firewall for malware detection [30], or considering a way for healing and backwashing alarmed transactions such as [31]. For the OR component, there is a room for speedup [32], and recognizing other attacks such as brute-force.

A possible direction may be to overload the existing OR component in optimizing the benefits of the blockchain financial aspects, based on integer programming for instance [33]. The ontology component may be extended [34]. For the Deep learning component, CapsNet [35] may be considered.

## References

1. Swan, M.: Blockchain: Blueprint for a New Economy. O'Reilly Media Inc, USA (2015)
2. Luu, L., et al.: Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM (2016)
3. Atzei, N., Massimo, B., Cimoli, T.: A survey of attacks on ethereum smart contracts (SoK). In: Principles of Security and Trust, pp. 164–186. Springer, Heidelberg (2017)
4. Xu, J.J.: Are blockchains immune to all malicious attacks? *Financ. Innov.* **2**(1), 25 (2016)
5. Frantz, C.K., Nowostawski, M.: From Institutions to Code: Towards Automated Generation of Smart Contracts (2016)
6. Noyes, C.: Bitav: Fast anti-malware by distributed blockchain consensus and feed-forward scanning. arXiv preprint [arXiv:1601.01405](https://arxiv.org/abs/1601.01405) (2016)
7. Hillier, F.S.: Introduction to Operations Research. McGraw-Hill Edu (2012)
8. Salgueiro, P., Abreu, S.: On using Constraints for Network Intrusion Detection. *INForum* (2010)
9. Salgueiro, P., et al.: Using constraints for intrusion detection: the NeMODE system. In: International Symposium on Practical Aspects of Declarative Languages. Springer, Heidelberg (2011)
10. Salgueiro, P.D., Abreu, S.P.: A DSL for intrusion detection based on constraint programming. In: Proceedings of the 3rd International Conference on Security of Information and Networks. ACM (2010)
11. Hassanzadeh, A., et al.: PRIDE: practical intrusion detection in resource constrained wireless mesh networks. In: International Conference on Information and Communications Security. Springer, Cham (2013)
12. Song, H., Lockwood, J.W.: Efficient packet classification for network intrusion detection using FPGA. In: Proceedings of the 2005 ACM/SIGDA 13th International Symposium on Field-Programmable Gate Arrays. ACM (2005)
13. Roesch, M.: Snort: lightweight intrusion detection for networks. In: *Lisa*, vol. 99, no. 1 (1999)
14. Huang, H.-D., et al.: Ontology-based intelligent system for malware behavioral analysis. In: IEEE International Conference on Fuzzy Systems (FUZZ) 2010. IEEE (2010)

15. Luh, R., et al.: Semantics-aware detection of targeted attacks: a survey. *J. Comput. Virol. Hacking Tech.* **13**(1), 47–85 (2017)
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436 (2015)
17. Weng, J.-S., et al.: DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain-based Incentive. *Cryptology ePrint Archive, Report 2018/679* (2018). <https://eprint.iacr.org/2018/679>
18. Huang, T.H.-D.: Hunting the Ethereum Smart Contract: Color-inspired Inspection of Potential Attacks. *arXiv preprint arXiv:1807.01868* (2018)
19. Dannen, Chris: *Introducing Ethereum and Solidity*. Apress, Berkeley (2017)
20. Metwally, O.: On the economics of knowledge creation and sharing. *arXiv preprint arXiv:1709.07390* (2017)
21. Delmolino, K., et al.: A programmer’s guide to ethereum and serpent (2015). [https://mc2-umd.github.io/ethereumlab/docs/serpent\\_tutorial.pdf](https://mc2-umd.github.io/ethereumlab/docs/serpent_tutorial.pdf). Accessed 23 Oct 2018
22. Jacobson, V., Leres, C., McCanne, S.: TCPDUMP public repository (2003). <http://www.tcpdump.org>. Accessed 23 Oct 2018
23. Eddy, W.M.: Syn flood attack. In: *Encyclopedia of Cryptography and Security*, pp. 1273–1274. Springer, Boston (2011)
24. Ornaghi, A., Valleri, M.: Man in the middle attacks. In: *Blackhat Conference Europe* (2003)
25. Noy, N.F., et al.: Creating semantic web contents with protege-2000. *IEEE Intell. Syst.* **16**(2), 60–71 (2001)
26. O’Gorman, G., McDonald, G.: *Ransomware: A Growing Menace*. Symantec Corporation (2012)
27. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: *European Conference on Computer Vision*. Springer, Cham (2014)
28. pypider.org. Accessed 23 Oct 2018
29. <http://mit.twman.org/TonTon-Hsien-De-Huang/research/deeplearning/R2D2>. Accessed 23 Oct 2018
30. Raje, S., et al.: Decentralized firewall for malware detection. In: *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*. IEEE (2017)
31. Continella, A., et al.: ShieldFS: a self-healing, ransomware-aware filesystem. In: *Proceedings of the 32nd Annual Conference on Computer Security Applications*. ACM (2016)
32. Schulte, C., Stuckey, P.J.: Speeding up constraint propagation. In: *International Conference on Principles and Practice of Constraint Programming*. Springer, Heidelberg (2004)
33. Mitchell, S., O’Sullivan, M., Dunning, I.: PuLP: A Linear Programming Toolkit for Python (2011)
34. Mundie, D.A., McIntire, D.M.: An ontology for malware analysis. In: *Eighth International Conference on Availability, Reliability and Security (ARES)*, 2013. IEEE (2013)
35. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: *Advances in Neural Information Processing Systems* (2017)