Gambling Detection using Smart Contract Flow Analysis

Daniel Chang danielchang@cmu.edu
Zhiyi Huang zhiyih@andrew.cmu.edu
Yichun Li yichunli@andrew.cmu.edu

1 Abstract

Modern cryptocurrencies provide far more functionality and capability that a typical fiat currency does not through decentralized applications (Dapps). Examples of these functionalities range from games to social networks to insurance applications. One of the more sensitive and legally questionable applications of cryptocurrencies is gambling. In this report we focus on how gambling is implemented using smart contracts and how we can detect gambling through flow analysis classification. We were able to achieve around 90% accuracy with deep neural network models.

2 Introduction

The rise of online gambling is pretty synonymous with the rise of the world wide web in terms of time and popularity. Laws were passed regarding online gambling that date back to the mid-1990s. Top trends driving the global gambling market include increasing availability of cash alternatives (cryptocurrency), growing popularity of online sports betting among younger audience (decreased average age from 45 to 38), increased penetration of international credit and debit card, and shift in consumer gambling habits (easier access to the internet on mobile devices). Gambling is a heavily regulated pastime and the rise of gambling applications based on cryptocurrencies undermine the efforts of lawmakers through pseudo anonymity and the current lack of jurisdiction.

Smart contracts are playing a crucial role in the changes in the market since the evolution of Ethereum allows for complex applications to be implemented, deployed and utilized on the blockchain. The Ethereum blockchain currently hosts a huge variety of applications in fields of games, finance and governance. Gambling smart contracts take a huge share in the mix as well. Online players are able to access and interact with these smart contracts to place and settle bets in many different forms. The flexibility and the functionality of smart contracts also allow the transactions to be conducted in forms of either fiat or cryptocurrencies.

The nature of gambling smart contracts ought to be restricted under legal contexts since some of the contracts are fraudulent schemes as they either are not inherently implemented in a fair and random fashion or they favor one party over the other. The majority of the states in the United States neither legalized nor forbidden online gambling activities for ordinary players under the Unlawful Internet Gambling Enforcement Act. This law mainly targets banks and online exchanges that process payments

related to gambling activities. Additionally, different states and countries have varying degrees of regulation and control of cryptocurrencies in general, which ranges from ignoring, all the way to bans or complete integration, with monitoring, recommendation, and guidance(whether anti-money laundering laws apply, and how tax should be applied) in between [1].

There are additional use cases of gambling detection systems outside of law enforcement. Many banks are already implementing gambling detection and blocking systems as a service to block incoming transactions related to gambling activities to help customers with gambling addictions [2]. It would be a useful extension to cryptocurrency exchanges as a feature.

The motivation behind the project is to automate the detection of the smart contract types and expedite the legislative efforts to regulate cryptocurrency-based gambling and other illegal activities such as money laundering using machine learning techniques. There is no prior work that we know regarding neural network classifiers that handle gambling contracts on the Ethereum blockchain.

3 Method

We found two types of detection methods for gambling contracts, white-box and black-box. White-box detection is when we have complete information about the gambling contract. It is when we know the function structures, parameters used for calling each function, and can track inner function calls directly. This has been researched extensively and is not our focus for this project [3][4]. Additionally, not all contracts have this information available. Only contracts that are verified by a trusted third party have this information. Specifically, the Application Binary Interface(abi) which is describing the function interfaces is missing for many contracts. Only having access to the contracts' bytecode is not helpful for our detection.

We would like to instead focus on the black-box detection method where we have no information of the inner workings of each contract and only rely on inputs and output transactions publicly available on the blockchain. Specific information that we are tracking includes the transaction value which is the msg.value field on each contract transaction, internal transaction value which is the value used for internal message calls triggered by regular function calls and are recorded separately from normal transaction value, and the token transfer value which provides transactions a method of exchanging tokens built on the Ethereum blockchain.

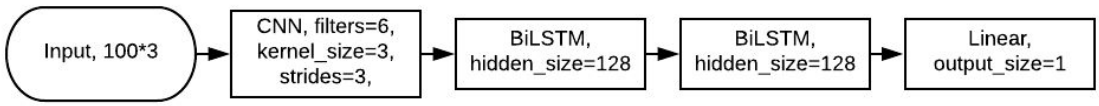
We used Etherscan's api to capture all three of these values [5]. We captured 1000 transactions from 223 contracts, of which 72 are gambling, 151 are non-gambling from categories like games, exchanges, social, governance, etc. Since the number of contracts we obtained is fairly small and could negatively affect the model performance, we split the 1000 transactions of each contract, represented by the triplet (transaction value, internal transaction value, token transfer value), into 10 samples of 100 transactions each.

The neural network that we built for the task consists of 5 layers in total, incorporating CNN and RNN. The first layer is the input layer and it takes in 100 sequences of triplets in chronological order.

The second layer is a 1-dimensional Convolutional Neural Network(CNN) layer. It uses convolution to summarize the information among neighboring transactions. This is a very common technique in machine learning. Convolutions are often applied in image and sequence processing to obtain relationships in the data across dimensions. In our case, applying it to the input time series better summarizes the data temporally.

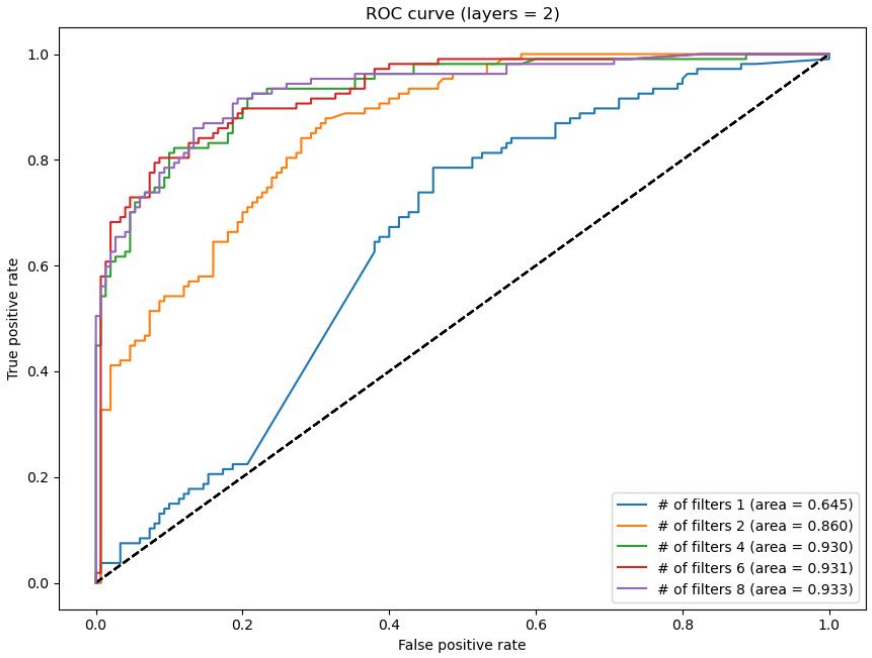
The third and fourth layers are Long Short-Term Memory(LSTM) layers. LSTM is another mechanism in machine learning that deals with sequences. It is good at learning and "memorizing" the relationships between things that are far apart from each other in the data, hence the name "Long Short-Term Memory". We need to use these LSTM layers for the ability to learn patterns across this long time gap. Bidirectional LSTM(bi-LSTM) learns these relationships from both directions: one way from the beginning of the data to the end, and the other way in reverse.

The final layer is the output layer with size 1. The layer uses the sigmoid activation function, which outputs the label 0, classifying the contract as non-gambling, or label 1, gambling.

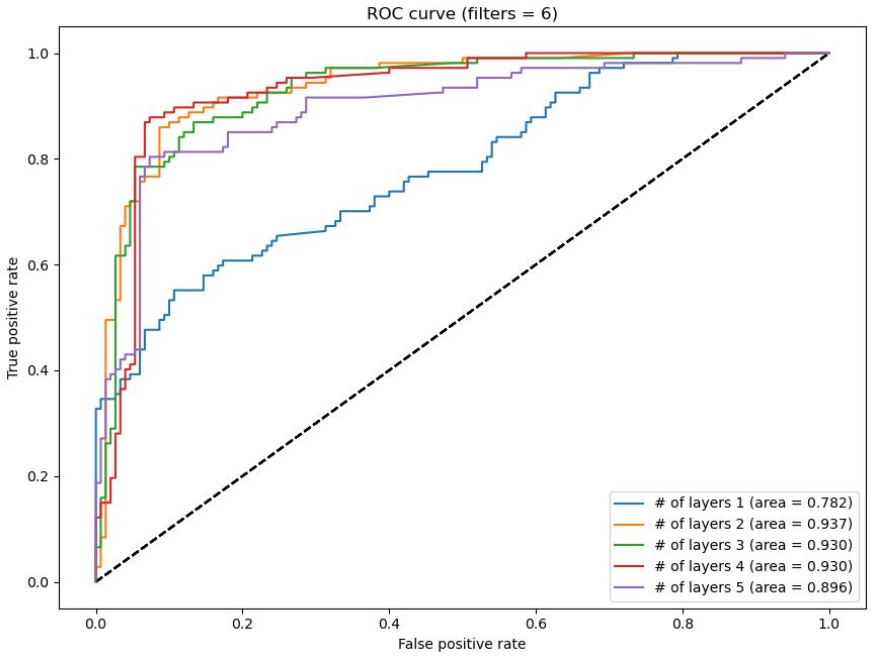


4 Results

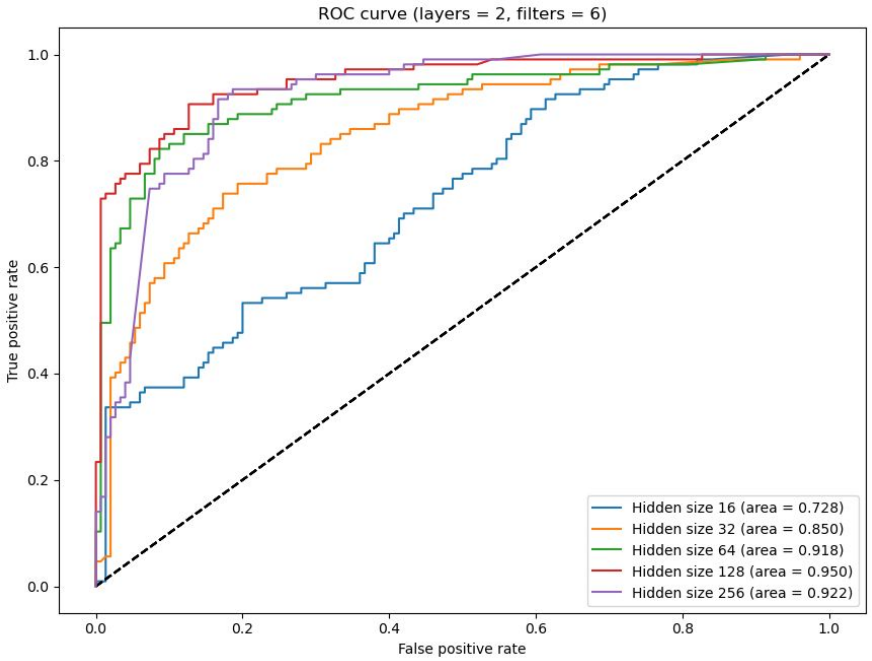
Using a common model structure and setting the number of epochs trained to 25, we tested multiple configurations of our model to find the most optimal one. We tested multiple filters using AUC_ROC as the metric to evaluate which of the configurations is the best.



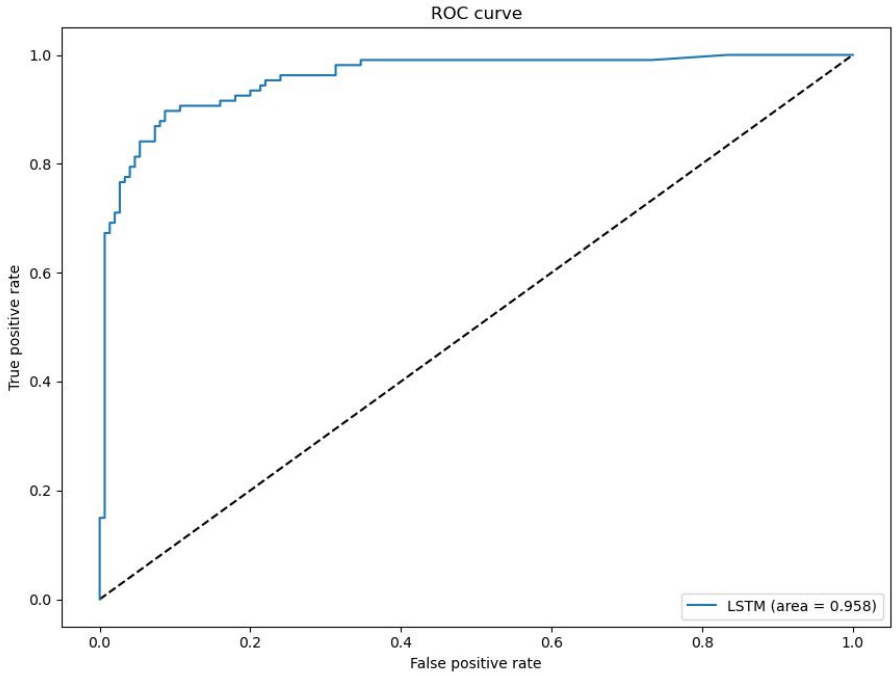
We also tested the number of bi-LSTM layers showing that 2 layers is enough for this classification problem.



We varied the LSTM hidden size and found 128 to be giving the best performance.



The current best configuration has approximately 90% accuracy on the test set averaged over 5 runs and the ROC curve is shown below.



5 Conclusion

In conclusion, we were able to successfully detect whether a smart contract is gambling or not with around 90% accuracy using our multi layer neural network. The data we used were all publicly available on the blockchain making it fit perfectly into a black-box detection method.

For future work, there are a few things that could be improved to make a better model:

1. More complex model. Using other machine learning mechanisms such as transformers could enable us to handle longer sequences and achieve better results.
2. More data. In our dataset, there are many contracts with small amounts of transactions. Given more contract transaction data, the model could give better results and be more generalizable.
3. Include more information from the transactions. One hypothesis we had was that gas and/or transaction fees used could be different for gambling vs non-gambling contracts. Function calls could be more involved for one type versus the other.

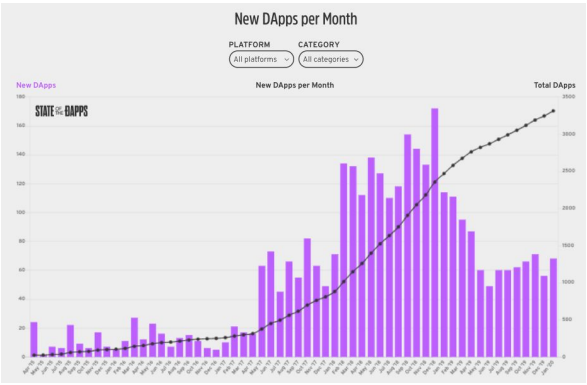
Our model is obviously not fool-proof and there are ways that gambling contracts could avoid from being detected. The model basically picks up patterns in the transaction sequences such as non-zero values, random values etc,. The contracts could apply tricks such as planting dud transactions or mimic non-gambling transaction patterns to get around our model if put into production. Currently, since there

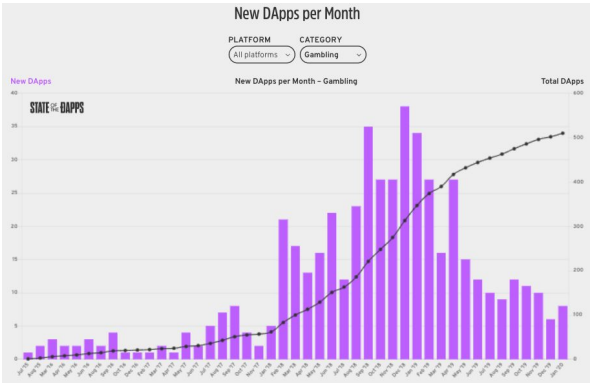
are no strong regulations on such gambling services as of this paper, and thus there are no attempts of deceiving detection systems. However, we presume they will be more popular if regulations catch up.

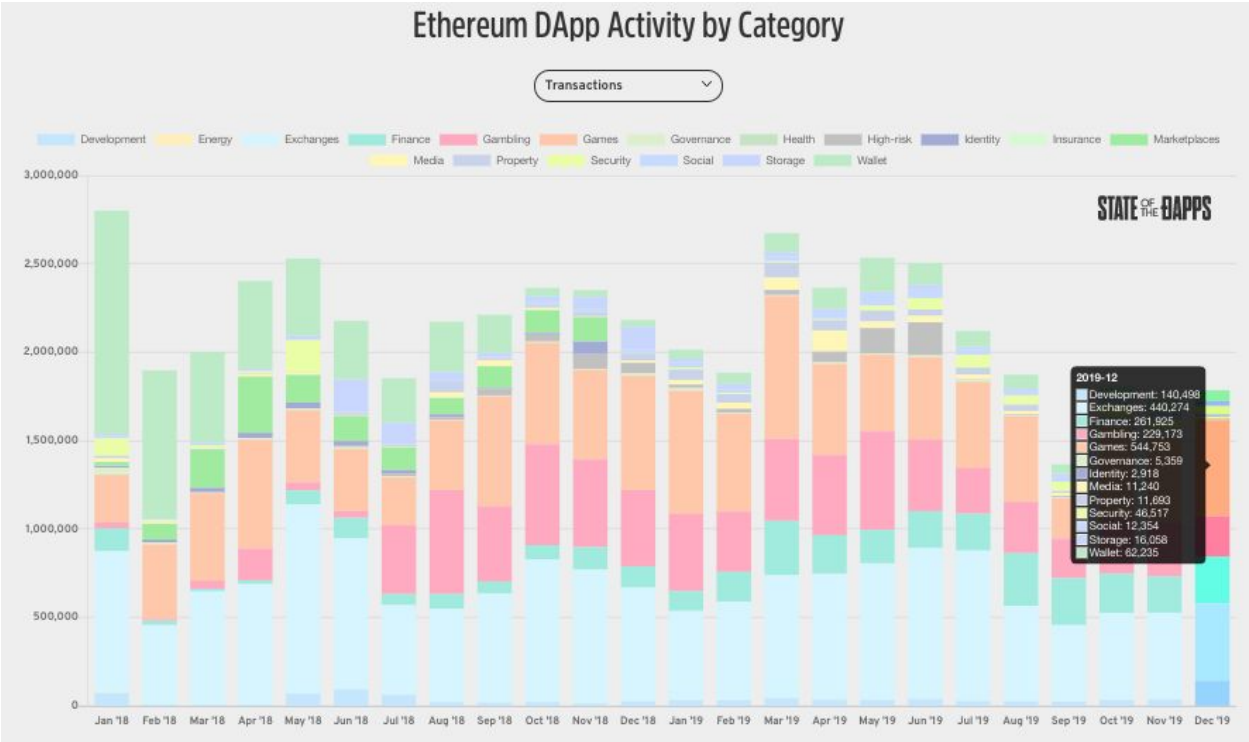
One benefit of the neural network approach to black box flow analysis is that the model can be easily adapted to other cryptocurrencies. Since there is no understanding of what the inputs are, as long as the data supports similar inputs, the model can be easily transferred to other domains.

6 Appendix

6.1: State of the DApps Charts [6]







6.2: Popular Gambling Systems

Types:

1. No Loss Lottery
    - Players purchase a token that is associated with staking or is sponsored by some corporation. The interest earned through that token is then randomly distributed to one of the players.
2. Short Term Options Trading
    - Short Term Options Trading contracts are gambling applications based on prices of cryptocurrencies. These contracts often involve the process of users betting on options with regards to the real time market prices of chosen cryptocurrencies within a certain time interval
3. Sports gambling
    - Very similar to traditional sports betting sites but provides several benefits through the usage of smart contracts such as: No signup required which ensures better anonymity, peer to peer betting removes the need of the house and thus reduces operation fees, peer to peer betting also removes the betting limit as long as two parties agree, allows bets wagering on something to not happen, can trade in-game for unlimited number of time prior to settlement.
4. Casino games
    - Implements casino gambling games using the block hash to create pseudo-randomness. All of the games are implemented based on the modulus of the block hash and a predetermined value in which the player wins if the specific modulus lies within a range of values depending on the type of game played.
5. Player vs player
    - The objective of the game is to guess correctly an action that your opponent takes or you could also opt to be a game host and have other players guess what action you took. You have a 50% chance to win no matter how many rounds you decide to play.
6. Lottery
    - Concept is the same as traditional lottery systems.

7 References

1. Lansky, J., Possible State Approaches to Cryptocurrencies. Signal Integrity Journal http://si-journal.org/index.php/JSI/article/viewFile/335/325

2. Griffiths, M., Hot topics: Gambling blocking apps, loot boxes, and ‘crypto-trading addiction’ http://irep.ntu.ac.uk/id/eprint/34067/1/11535_Griffiths.pdf

3. Albert, E., Gordillo, P., Livshits, B., Rubio, A., Sergey, I., EthIR: A Framework for High-Level Analysis of Ethereum Bytecode, https://link.springer.com/chapter/10.1007/978-3-030-01090-4_30

4. Grishchenko, I., Maffei, M., Schneidewind, C., Foundations and Tools for the Static Analysis of Ethereum Smart Contracts, https://link.springer.com/chapter/10.1007/978-3-319-96145-3_4

5. https://etherscan.io/apis

6. https://www.stateofthedapps.com/rankings/category/gambling