Blockchain-based Smart P2P Lending using Neural Networks

Bhaumik Choksi

KJ Somaiya College of
Engineering
Mumbai, India

Alisha Sawant

KJ Somaiya College of
Engineering
Mumbai, India

Sai Subhasree Pakina

KJ Somaiya College of
Engineering
Mumbai, India

ABSTRACT

Over the past decade, there has been an exponential growth in the number and scale of online lending and crowdfunding platforms. However, these platforms lack a reliable and transparent metric to predict the credit-worthiness of an applicant. They also have a single point of failure and are vulnerable to certain security issues. This paper proposes a Blockchain-based decentralized lending platform that uses deep learning to predict the risk associated with an applicant. The paper also discusses how such a system can be implemented and deployed. The experimental results show how ensemble training can help lower the bias of individual neural networks and provide better predictions for this use case.

General Terms

Machine Learning, Artificial Intelligence, Finance, Lending

Keywords

Deep Learning, P2P Lending, Credit-risk, Blockchain, Decentralized, Neural Networks

1. INTRODUCTION

Blockchain-based cryptocurrencies have become hugely popular recently. They allow for seamless transactions all over the world with minimal charges and do not involve any financial institution, such as a bank. Furthermore, it is practically impossible to hack into, and therefore, virtually immutable.

Given this immense popularity of the Blockchain technology, an increasing amount of money is being invested in decentralized P2P systems. Some P2P platforms have thrived, and are being used increasingly by startups and small businesses, whereas some have struggled due to diminishing participation or security flaws. Today, banks and large traditional financial institutions still dominate the lending scenario.

Most banks have a solid and accurate means of verifying a customer's identity and also to determine the creditworthiness of an applicant. This is something most online platforms lack. Lenders in P2P lending bear the risk of loan default because of the unsecured nature of the loans [1]. A manual credit risk assessment requires a high level of expertise, and still cannot eliminate human error [1]. Artificial neural networks have emerged as a superior and robust technique to make accurate predictions, even for data that may seem hard to visualize. It is possible to used neural networks to process the available customer data and attempt to make accurate predictions for future customers. In order to minimize risk and to choose where the money should be granted, a critical evaluation of loan applications should be carried out to come to a reliable and effective decision [2]. This paper focuses on designing a decentralized and peer-to-peer system of credit transfer and lending. Such a system would be immune to fluctuations in currency prices, and not controlled by any central authority. Our proposed model is built on the Ethereum network. Smart contracts are used to implement the business logic. This means that anyone with enough credits can participate in the lending system, and perform transactions with little to no overhead, as opposed to the traditional banking system.

The organization of this paper is as follows. Section 1 is the introduction. Section 2 describes relevant background knowledge and concepts related to P2P lending, Blockchain, and Artificial Neural Networks. In section 3, the architecture of the proposed system is explained in detail. Section 4 contains the design of the neural networks used and details about the chosen training algorithms. Section 5 describes the datasets used for testing our hypothesis, the features that are taken into consideration for making predictions, and details about the training process. The results of our tests are presented in section 6. Finally, in section 7, the conclusions are presented, along with further lines of work in section 8.

2. LITERATURE SURVEY

There have been several studies and experiments on managing credit risk. Customer data is a highly valuable asset and can be used to find many interesting patterns and details about the customer's behavior. Credit scoring is defined as a statistical method that is used to predict the probability that a loan applicant will default or become delinquent [3]. Credit scoring helps to increase the speed and consistency of the loan application process and allows the automation of the lending process [3].

Ghatge et al. [3] showed how an ensemble neural network could be trained on a selected feature set in order to predict credit default. Alaraj et al. [2] showed how various optimization techniques could be used in order to enhance the accuracy of such a model. They used techniques like bagging and data normalization to improve the quality of predictions.

Banks in India use the CIBIL score to determine the worthiness of a loan applicant. But, in P2P systems loans are unsecured and it is difficult to classify the applicants as good or bad. Thus, the substantial risk of default prioritizes determining the credit risk associated with every loan applicant. P2P business can generate profits only when the investors gain high return on investment and the borrowers are satisfied with the interest rates.

An ideal way to implement such a P2P system would be by leveraging the Blockchain technology. It uses a digitally distributed ledger that logs all the transactions that have taken place. Every node must synchronize itself with the current state of the Blockchain. Therefore, the failure of one node will not have any significant impact on the system. It also eliminates the need for any central authority. Since every

block in the Blockchain is cryptographically verified, it is computationally infeasible to alter any part of the Blockchain.

Blockchain-based platforms have also become a popular way to raise capital. Startups have started using Initial Coin Offerings (ICOs) in order to raise funds for their ventures. Such a system helps them bypass banks. If the money raised does not satisfy the minimum requirement of the firm, the money is returned to the investors and the ICO is deemed to be unsuccessful. If the requirements are met within the specified timeframe, the money raised is utilized in order to either initiate the new scheme or to complete it.

3. SYSTEM ARCHITECTURE

The proposed system utilizes smart contracts on the Ethereum network to create a decentralized lending platform. The architecture is shown in Figure 1. The entire architecture can be divided into three levels – the smart contract, the Web3.py interfacing module, and the Python backend containing the neural network prediction module.
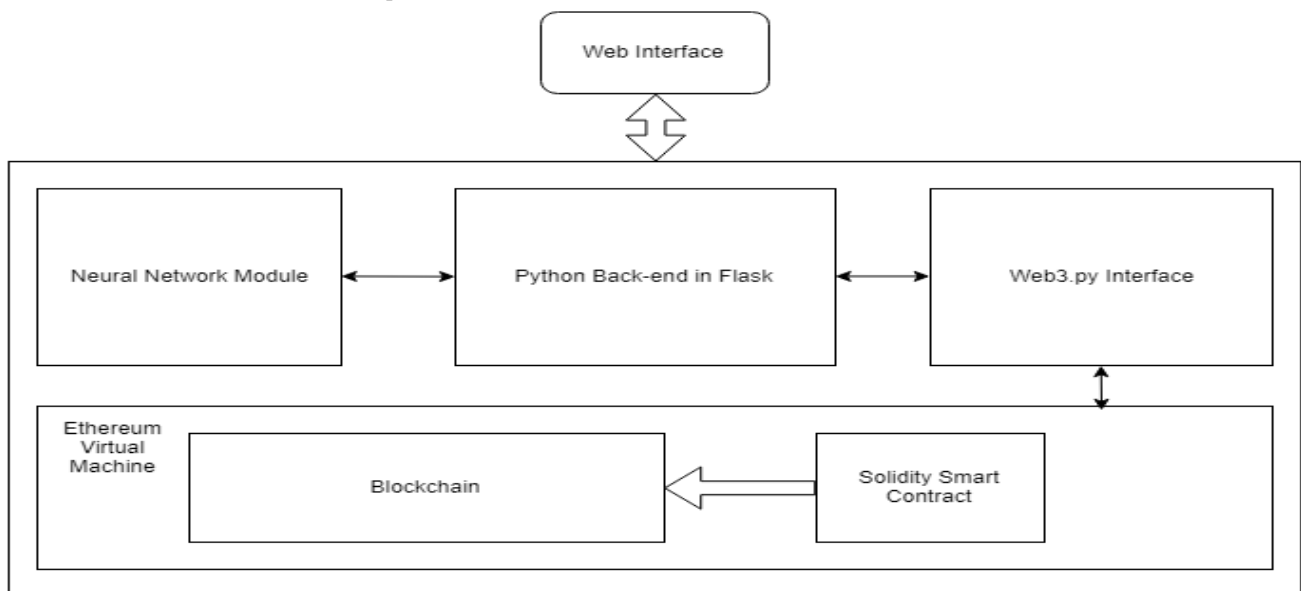
3.1 Blockchain

Fig 1: Architecture of the proposed system

A Blockchain is a distributed and decentralized ledger hosted on a global network of nodes. A Blockchain stores transactions in the form of a cryptographically signed block. Blocks are mined and added to the Blockchain by miners, who perform computationally expensive operations to verify the authenticity of the block. Nodes then use a consensus mechanism to determine which block gets added to the Blockchain. Since the hash of each block depends on the hash of the previous block, altering any block would require the attacker to re-calculate the hash of every block that came after the modified block, which is very difficult when the number of nodes is large. Every transaction is witnessed by all the users thus leading to faultless delivery. Therefore, the Blockchain provides a solid foundation for developing decentralized applications.

3.2 Smart Contracts

Smart contracts on the Ethereum network allow developers to deploy program logic onto the Blockchain. Usually, the Blockchain only stores transactions, but with the help of smart contracts, the developers can also store their program on the Blockchain. This makes the code immutable, and globally accessible. The state of the contract can be changed using function calls. In the proposed system, smart contracts are used to store all the information and the functions needed to modify this information over time. The contract is written in Solidity. It is then compiled and deployed onto the Ethereum Blockchain, and the address of the contract is stored. Once it is deployed, every node on the network can access the contract using this address. Our smart contract helps store user

identity, user type, account information, loan applicant information and has functions that facilitate application creation, approval, and repayment.

3.3 Web3 Interface

In order to communicate with the smart contract, the proposed system uses the open-source Web3.py framework. It allows the client node to execute transactions and interface with the functions written in the smart contract. This creates a layer of abstraction between the Python code and the Blockchain. It is possible to implement certain functionalities at the interface level, so as to minimize the computations that take place on the smart contract. The system uses many helper functions in this layer in order to reduce the computation time. User authentication is also implemented in this layer.

3.4 Python Backend

The Python backend communicates with all the other modules and is present on every node that runs this application. Flask is an open source Python library that allows users to write backend scripts. This module communicates with the neural network module to make predictions, and also sends and receives data from the Web3 interface. The major role of this module is to render the web-based user interface for the

clients. Input validation, views, and forms are handled by this module.

4. PREDICTING LOAN DEFAULT

Researchers have been developing scoring models based on new techniques, such as hybridization and ensemble

techniques which show higher accuracy than individual models [2]. In order to identify the credit risk associated with an application, the proposed system uses a neural network based prediction module. It consists of three feed-forward neural networks that are trained on different subsets of the data using different training algorithms.

4.1 Design of the neural networks

Each of the three feedforward neural networks contains 4 layers. The first layer consists of 200 neurons, followed by 2 layers of 500 neurons each. Finally, the output layer consists of 2 neurons, one for each category – risky or safe. The softmax activation function is used for the output layers, in order to facilitate multi-class classification. The magnitude of the output at each output neuron signifies the confidence level of the prediction associated with that class. Figure 3 shows the

design of each individual neural network. Figure 2 shows how three of these neural networks are combined to form an ensemble.

4.2 Training algorithms

The three neural networks are trained using separate training algorithms. The first network is trained using the Adam optimizer since this algorithm is computationally efficient and has low memory requirements. The second network is trained using RMSProp, which shares some of its strengths with the Adam optimizer. The third neural network is trained using Stochastic Gradient Descent since the algorithm tends to converge quickly in general. The categorical cross-entropy loss function is used to calculate the loss since multi-class classification is being performed.
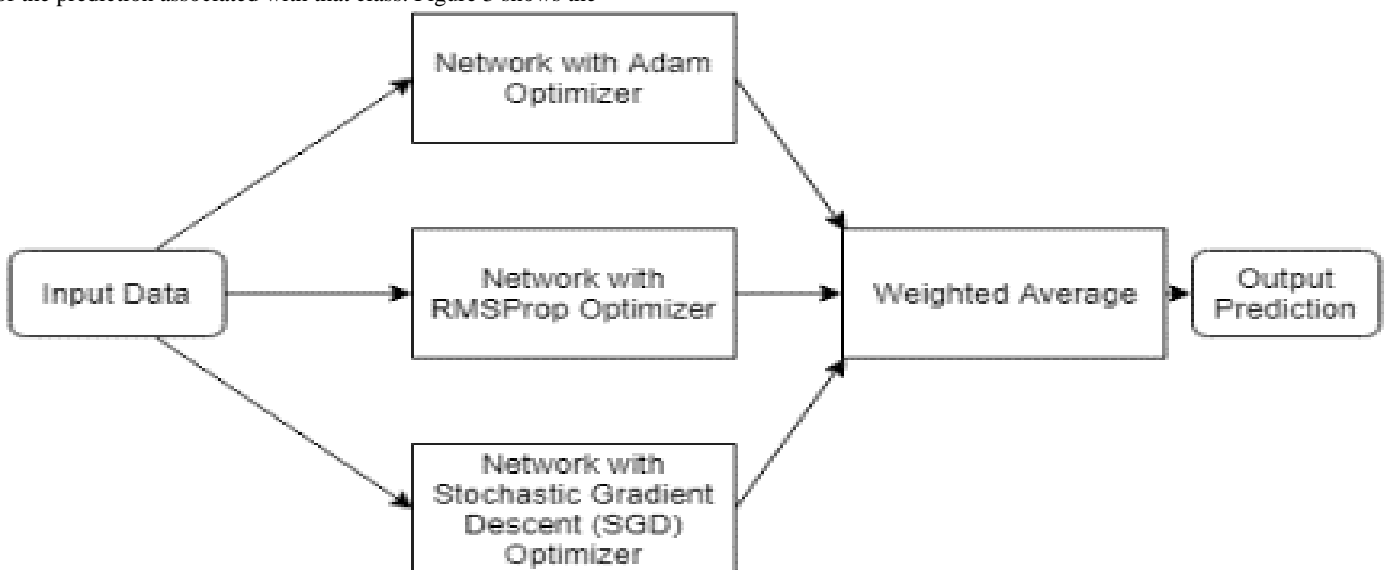

Fig 2: Neural Network Ensemble

5. METHODOLOGY AND EVALUATION

The proposed ensemble learning technique was tested on two public datasets - The Lending Loan Club dataset from Kaggle, and the Statlog German Credit Dataset from the UCI Machine Learning Repository. Both of these datasets are public, and have been used in previous research and experiments based on this topic. Both of these datasets include numerical attributes like loan amount, interest rate and annual income of the applicant, and also categorical attributes like gender and loan purpose. For the proposed model, we use the attributes present in the German Credit Dataset.
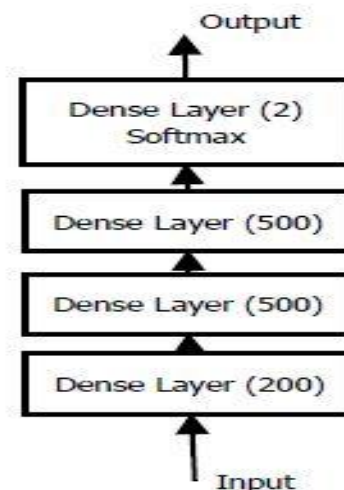

Fig 3: Design of each neural network

5.1 Data Description

The Lending Club dataset is obtained from the Lending Club platform, which is an online P2P lending platform. It consists of over 700,000 loan records and a total of 75 attributes.

Every tuple has a loan status associated with it, which is used as the output label.

The German Credit Dataset is relatively smaller, and contains 1000 instances. Each instance consists of 20 attributes – 7 numerical and 13 categorical, along with an output status which can either be 1 (Good) or 2 (Bad).

5.2 Data Preparation and Preprocessing

Since the given data can be noisy and inconsistent, it must be cleaned to enhance the accuracy of the model, and the quality of the predictions. Tuples with missing values are removed. Attributes with high correlation are also removed to reduce redundancy.

Categorical features and numerical features are pre-processed separately. Normalization is performed on the numerical features. Categorical features are mapped to numerical labels, and these numerical labels are one-hot encoded to get a vector. The output labels are also one-hot encoded. One-hot encoding helps calculate the loss more accurately and greatly increases performance.

5.3 Training

The model used for testing our hypothesis was implemented in Python using the Keras framework. Each of the three neural networks was trained independently for 30 epochs, with a batch size of 5.

An 80-20 split was performed to get the training data and the testing data, where 80% of the data was used for training and 20% was used for testing. The data was shuffled randomly before performing the split.

6. RESULTS

The results obtained after training were plotted using the Tensorboard utility. The plot of the training loss and training accuracy over the training epochs has been shown in Fig. 4 and Fig. 5 respectively.
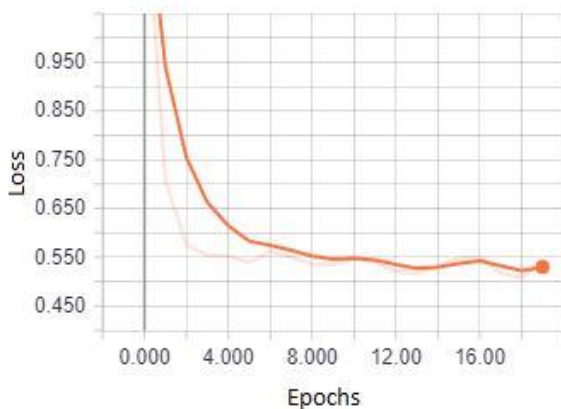

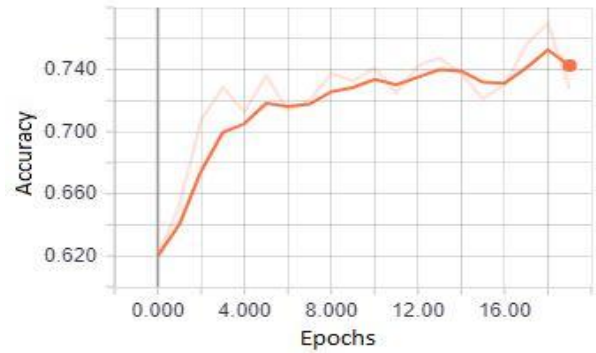**Fig 4: Training Loss**

Fig 5: Training Accuracy

The results show that each individual model was able to classify with an accuracy of at least 70%. The loss decreased for a few epochs and gradually became stable. Table 1 contains the percentage accuracy obtained after the training was complete.

Table 1. Training results for the proposed model

Optimizer	Training accuracy	Testing accuracy
Adam	72.5 %	72 %
RMSProp	74.5 %	74 %
SGD	74 %	74 %

Table 2. Training results from an earlier model in [4]

Optimizer	Accuracy	Mean Square Error
Levenberg–Marquardt	76 %	0.17
Scaled Conjugate Gradient	65 %	0.30
One Step Secant	55 %	0.23

7. CONCLUSION

The ensemble technique increases the accuracy of the model, as it is seen from the experiments. The proposed P2P lending model not only reduces the dependence of users on large financial institutions, but also allows the users to invest in various campaigns directly. It also provides a secure platform for these financial transactions by leveraging the Blockchain technology. The neural network module helps investors assess the credit risk of a given application so that they can make an informed decision while investing.

8. FUTURE SCOPE

Due to the infancy of the Ethereum platform and decentralized systems in general, several changes and optimizations can be made to the proposed model. Instead of having a single contract store all the information, a new contract can be deployed for every loan agreement, allowing for one-to-many and many-to-many contracts. The training and periodic retraining of the neural network can also be optimized further. Finally, new features can be experimented with to determine their effectiveness in the given use case.

10. REFERENCES

[1] Byanjankar, Ajay, Markku Heikkilä, and Jozsef Mezei. "Predicting credit risk in peer-to-peer lending: A neural network approach." Computational Intelligence, 2015 IEEE Symposium Series on. IEEE, 2015.

[2] Alaraj, Maher. Evaluating Consumer Loans Using Neural Networks Ensembles. ICMLEME, 2014.

[3] Ghatge, A.R. Ensemble Neural Network Strategy for Predicting Credit Default Evaluation. IJEIT, 2013.

[4] Amira Kamil , Ajith Abraham Modeling Consumer Loan Default Prediction Using Ensemble Neural Networks. ICCEEE,2013

[5] Jin, Yu, and Yudan Zhu. "A data-driven approach to predict default risk of loan for online Peer-to-Peer (P2P) lending." Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on. IEEE, 2015.

[6] De Filippi, Primavera. "Blockchain-based Crowdfunding: what impact on artistic production and art consumption?." (2015).

[7] Jacynycz, Viktor, et al. "Betfunding: A distributed bounty-based crowdfunding platform over ethereum." Distributed Computing and Artificial Intelligence, 13th International Conference. Springer, Cham, 2016.