Metaverse Credit Scoring System

Credit scoring and sharing model based on blockchain

Metavisa Team
hello@metavisa.com
www.metavisa.com
November 2021

Abstract

With the rapid development of decentralized finance, some lending protocols have seen flash credit attacks caused by lending protocols only evaluating collateral for users without practical evaluation of investors' credit risk. This paper draws on traditional credit scoring models such as the logistic regression model and the emerging standard model represented by machine learning, such as decision tree. MetaVisa Protocol conducts comprehensive processing and evaluation of data in various dimensions such as Credit history, On-chain behavior preference, Address activity level, Asset holdings & Portfolio, Address correlation. The logistic regression model, SVM type model, and decision tree type model are trained separately. After the model prediction, the metrics of model training efficiency and model prediction effect are evaluated together, and the single data mining model with better model evaluation is selected. And the XGBoost model and logistic regression model are Stacking model fusion to get the final model optimization form, which is contained for model training and prediction. The optimization of the credit scoring model with data mining technology for on-chain address data is completed. In addition, this paper proposes a blockchain-based, decentralized and trustworthy credit sharing model.

Background

This section opens the discussion of the missing issue of the Metaverse credit scoring system from the value of credit score in the traditional world and states the credit scoring method from through literature review, and details the application of data mining theory in the conventional credit scoring field.

The value of credit score in the traditional world

The history of credit relationships accompanies the development of the human social division of labor and transactions. As long as there are trading activities, there are contracts created, and thus individual credit exists. From the object of credit granting, credit can be divided into three categories: government credit, corporate credit, and personal credit. While government credit and corporate credit are generally reflected by organizational behavior, they are traced back to a collection of individual behaviors, the overall expression of multiple individual behaviors. Therefore, personal credit is closely related to all social and economic activities.

A personal credit score is an assessment of an individual's ability to fulfill economic contracts based on a comprehensive examination of the subjective and objective factors of the individual's environment.

At present, the purpose of scoring personal credit is usually threefold.

- Minimization of customer default rate. By identifying and classifying the credit and risk levels of customers, credit evaluation agencies are trying to exclude customers with fraudulent credit intentions and those who will not be able to complete their repayments on time and in quantity in the future, thus allocating limited resources to good customers, reducing the risk of default and making better lending decisions.
- Optimize customer management. Credit rating agencies rely on individual credit scores for effective customer management, including more refinement of target customer groups and thus targeted innovation and recommendation of new products or services; more effective prediction of dynamic changes in customer risk levels and consequent dynamic allocation of corresponding credit authority and credit levels, and timely and better management decisions to reduce alarming debt rates and debt losses, etc.
- Maximizing profits. Credit rating agencies evaluate and judge individual credit, reduce default rates, allocate resources more effectively, and optimize customer management, all to assess the yield and value generated by customers and ultimately achieve the goal of profit maximization for credit rating agencies.

The absence of a Metaverse credit scoring system

As the underlying infrastructure of Metaverse, Blockchain carries a large amount of interaction information between users and applications in Metaverse. Although this interaction information can be queried in the blockchain, it isn't easy to provide intuitive feedback on the credit situation of address owners because the data is vast, complex, and disordered. And users will protect their privacy by switching addresses, thus making it challenging to accumulate credit in one address.

However, the Metaverse, as an essential evolutionary direction for human civilization, is bound to generate a large number of applications, and credit scores can change to enhance the value created by applications for users. However, the credit scoring system is still missing in the Metaverse. The off-chain interaction data between different applications are not connected, so the credit system is not accessible. Therefore, establishing a unified credit scoring standard plays a crucial role in opening up credit ecological services.

The blockchain-based data desensitization method and the smart contract established on the blockchain can guarantee the fusion and analysis of multi-source data under the condition of metadata security, which effectively prevents information security issues in the existing data-sharing process. For example, individuals' social relationship and consumer financial data in the Internet platform can be confidently authorized to smart contracts for full credit assessment. As a result, users can get the credit score assessed by credit evaluation agencies for themselves. In this process, other users do not get the metadata of the user. Therefore, in Metaverse, the credit scoring based on the blockchain data can maximize the value of the data.

Research on Credit Scoring Methods

Around 1950, credit scoring models were gradually explored and developed. The emergence of credit cards and credit ratings such as home mortgages advanced the credit scoring research process. Initially, individual credit scoring methods were often evaluated by a sample of relevant experts based on their own rules of thumb. The study obtained several representative results of typical significance. Still, these methods have limitations, arbitrariness, and subjectivity. On top of this, personal credit scoring models based on various mathematical and statistical methods were mined. These methods are based on the knowledge of economic management and other disciplines to mine and organize individual samples' information and provide a comprehensive and integrated measure of individual behavior. Personal credit scoring models can portray the probability of an individual's default to the maximum extent possible by describing unique characteristics and reducing the misclassification rate of personal credit models.

In the beginning, credit scoring models were mainly used to analyze individual credit behavior by traditional methods to understand their general development. Later, many other fields of knowledge were incorporated into personal credit models, such as discriminant analysis and logistic regression, which gained wide recognition and respect. As a result, individual credit scoring methods are gradually evolving and upgrading and have fully incorporated statistical and computer knowledge.

Single credit scoring methods include statistics, non-parametric, operations research, artificial intelligence, etc. There are numerous research results on single credit scoring methods, and there have been many groundbreaking new developments in applications and methodological improvements in recent years.

Statistical methods

Statistical methods include distance measurement, attribute study of individual credit, among which discriminant analysis and cluster analysis are generally recognized. Discriminant Analysis (DA) was developed based on Fisher and has gained academic recognition. David Durand(1941) used discriminant analysis for individual behavior studies, focusing on the qualities of misclassification rate and its role in credit scoring. Discriminant analysis is based on a measure of the distance of a specific pair of sample attributes with small distances between like categories, and individual attributes are analyzed using single individual discriminant models and combined discriminant models. Hand argues that if sample bias is ignored when the sample variables approximately obey a normal distribution, it indicates that the sample itself can be more suitable for discriminant analysis. Hardy in 1985 applied Discriminant analysis to business, finance, and other fields and used it for comparative analysis of credit scores. However, it is worth noting that credit data modeling studies in which the above conditions cannot be met do not apply to the above methods.

Rosenberg explores the applicability of discriminant analysis and points out its contradictory results. This approach has a constraint that the information collected is often from customers who have already taken out loans. In contrast, data from those other customers are eliminated before discriminating, and discriminant analysis has deviated from its distance properties due to its characteristics. Many later models have evolved based on this model. FICO was developed to study in the context.

Regression analysis is one of the primary research methods, applying Darwin's genetic factors in the research context. The regression analysis method is based on primary research to analyze the factors influencing a variable, study the mechanism of the inner connection between variables, and study the dependent variable's dynamic change process through the analysis of the independent variable. Regression analysis takes complete account of the attributes of the variables, the internal structural relationships between the variables, and the advantages in credit scoring. Therefore, regression analysis has gained general acceptance, making the method more applicable. Orgler applied the technique of regression analysis to study the subordination of indicators. Fitzpatrick studied the relationship between the loan and credit assessment aspects and tested it empirically. By far, logistics has been the most commonly used, expanding the study of statistical methods.

Non-parametric methods

contains many components to understand the ways of individual credit scoring through statistical analysis of variables and empirical generalization. The Nearest Neighbors (NNs) method has strong applicability in dealing with classification issues, which was proposed by foreign scholars in the 1950s and has since been recognized by the American banking community and fully applied to individual credit assessment. Henley used it in consumer credit assessment to compare methods analysis, provide a detailed and in-depth analysis of the nearest neighbor method's distance analysis principle, and analyze its advantages and disadvantages in handling data. Yeh dissected the features of the nearest neighbor method, which does not require pre-simulation exercises, the information can be processed, and it shows better scoring results in individual credit rating.

Scholars around the 1980s proposed the decision tree method (DTM), and it is mainly based on the characteristics of variables for classification processing and measurement. The decision tree method recombines data to classify variable attributes as a whole, with relatively small variability between subclasses, until the training sample is satisfied by the constraints and the classification is complete. Makowski applied the decision tree method to personal credit scoring and achieved good results. Olaru developed the decision tree method by combining the Fuzzy method with the decision tree method and established SDT (soft decision trees) model and applied it to credit scoring, which improved the classification accuracy of the decision tree method. Nie combined logistic regression with the decision tree method and established a combined logistic regression-decision tree-based individual credit scoring model, which combined the two methods' advantages and reduced the misclassification rate of personal credit scores. It has been suggested that the process of applying decision trees to individual credit scoring is to classify applicants into N subclasses according to different criteria, discriminate the N subclasses separately, and determine the possible future default of customers in each subclass. The decision tree approach provides a new way of thinking to perform classification, is easy to understand and implement, and can directly characterize the data. However, its ability to handle continuous numerical variables is weak, and the classification accuracy still needs to be improved when applied to individual credit scoring.

Bayesian networks were developed based on the probability theory of mathematics. The method is based on the relevant theory to obtain the scoring results, analyze the attribute factors of individuals in general, and explore the characteristics of individual variables. Many scholars have recognized the method in terms of utility and accuracy of credit scores, such as Baesens and Migueis, who have tested it empirically through credit samples.

The clustering method is more widely used in personal credit scoring, based on different clustering rules to analyze the indicator variables in personal credit assessment in terms of weighting, and then classify the study by clustering rules. This method can improve the accuracy of scoring and be more targeted to solve the sample bias issue. The FCM method has some advantages in solving the individual credit scoring issue due to its assumptions and application restrictions. This method optimizes data mining rules, has advantages in solving economic problems, and several other problems in the application of credit scoring. In contrast, this method has few assumptions and can be used in combination with other methods.

Linear programming method

This method has an essential role in individual credit assessment, supported by mathematical theory (LinearProgramming, LP. Mangasarian pioneered the operational analysis in individual credit assessment. However, it is only since the 1980s that this approach has gained widespread attention. Linear programming models generalize excellent and bad samples according to specific attribute characteristics by setting a threshold value. The good sample points are in the upper half of the critical value curve, and the bad examples are in the lower half of the critical curve. The difference between linear programming and other methods has been the focus of scholars. Nath pointed out through a study that statistical methods are superior to those of linear programming. Shi pointed out that multi-criteria linear programming methods are superior to statistical methods when applied to individual credit scoring. Ic incorporated the fuzzy methods approach into linear programming by reworking the model and analysis so that the misclassification rate of the sample of individual credit assessment is reduced and the overall risk prevention mechanism of commercial banks is improved.

Artificial intelligence methods

Statistical knowledge was first applied to individual credit assessments. However, the late 20th century saw significant advances in credit assessment technology, especially the development of artificial intelligence technology. The application of artificial intelligence methods to individual credit scores also showed diverse characteristics in processing samples.

After 1970, artificial intelligence entered an era of booming development, and the popularity of computers also contributed to the development of the method. One of the better results has been achieved by ExpertSystem, which processes the information of experts in classification processing by discriminating the intrinsic links between variables. In personal credit classification processing, expert systems are not based on the analysis of empirical information but the neural system approach to classification.

Neural networks (Artificial Neural Networks, ANNs) have achieved initial results and occupy an essential place in the field. West applied this method for credit rating analysis and the establishment of individual credit systems. Abdou used neural network methods and achieved long-term progress. Because artificial neural networks have their methodological advantages. The permutations of variables and parallel changes in data allow for adequate running and processing of samples. Furthermore, the method is constantly updating and evolving the method itself. As a result, neural networks have shown more significant advantages in solving personal credit processing.

The basic principle of the support vector machine is the analysis of the sample space and the processing of the surface fit. The distance to the surface is statistically analyzed. The comparative analysis of the distance between the sample points on the surface, i.e., the greater the individual vertical distance of the sample points, the higher the level of optimization of the sample set, and the surface of the sample points is the practical embodiment of those critical scatter points. The characteristics of the support vector machine itself can be combined with several methods to solve personal credit scoring issues. Minghui Jiang first tried to use the method in combination with regression analysis, GA, PSO, and other optimization to reduce the misclassification rate and improve the risk resistance.

Genetic algorithm is the process of integrating computer science and biological science methodological mechanisms with each other, and superiority genetics is applied to the sample selection process to generate an adaptation function to find the sample with the highest adaptation as the optimal solution. The applied mechanism of the method has high applicability in solving individual credit scoring issues, but studies have shown that the computational process of the method is relatively complex. Scholars have confirmed that the integration of support vector machine methods with other methods is also of high academic value and empirical significance. Academic research on a single model has found that a single model approach has obvious advantages and, at the same time, limitations in conducting the application process of individual credit scoring. Statistical methods are able to discover the regularity of data well and have strong explanations; however, their application prerequisites often require the data to obey a special distribution due to the excessive data requirements, and the realistic applicability is challenged. The data processing method of artificial intelligence reduces the requirement of variable characteristics and improves the prediction accuracy of individual credit; however, the transparency of the artificial intelligence method is not high, and it is difficult to reflect the economic meaning of variables. Under the constraints of strict assumptions, complicated computational rules, and restrictions on moderating variables, most studies of single models focus mostly on the accuracy of the models and not enough on the explanatory studies of the models. Therefore, people have started to use combinatorial models for individual credit scoring using the characteristics and complementary features of different methods.

Combined credit scoring methods

Combinatorial credit scoring methods are combinations of two or more methods that enhance the applicability of a single model by applying the algorithmic rules of combined weighted equilibrium. The first to propose a combined credit scoring method was Bates and Granger. They systematized the rules of the combined method in 1969 and thereafter, Cleman focused on the study of the combined method and discussed and analyzed the predictive function of the method in depth. Based on this, more combination rules were applied, including average method, winning method, optimal method, regression method, difference method, etc. Scholars such as Kuncheva explored combination rules and mined combination methods with greater applicability, while Siami and others used combination scoring methods to conduct an in-depth analysis of sample and indicator variables.

The research on combined credit scoring models has seen an increasing number of combined rules and models emerge, and scholars have used the complementary strengths between models to improve the accuracy of scoring predictions. As a result, scholars have tried to combine and integrate more than two or more methods for optimization, and proposed individual credit scoring models based on integration learning methods. The difference between the combined models and the integrated models is that: the single models that are combined are often two or three, but the single models used for integration exceed the size of the combined models in number; secondly, the single models that are combined are selected manually based on the characteristics and applicability conditions of the models and ensure the complementarity before each other, but the results show that sometimes the models are seen from the characteristics of their algorithms complementarity, but the practical application is not good. In comparison, the integrated algorithm has a high degree of automation in model selection, and the integrated approach performs automatic screening by prediction results, and the integrated model has higher applicability characteristics when performing individual credit scoring.

Application of data mining in the traditional credit evaluation field

By analyzing the research related to data mining technology in assessing credit risk in traditional finance and Internet finance, it is found that data mining technology can be applied in the financial field and its effect on credit risk assessment is better than the traditional credit scorecard system. Especially in the Internet finance field, influenced by the big data environment in the Internet finance field, there is more room for the development of data mining technology in assessing the credit risk of P2P individuals compared with traditional statistical analysis methods.

In the traditional financial field, the credit risk assessment of borrowers by traditional financial institutions such as commercial banks is generally carried out through the credit score card system. The so-called credit score card system refers to the fact that financial institutions, when reviewing loan or credit card applications for borrowers, generally focus on analyzing the basic information and credit history of borrowers, dividing these two assessment indicators into several small assessment indicators, according to the borrower information on each The financial institution can evaluate the credit risk of the borrower based on the total credit score, which is the credit score card system. With the continuous development of big data and data mining technology, many scholars began to apply data mining techniques to the traditional financial field. Bee (2012) was able to effectively assess the credit risk of borrowers by constructing a logistic regression model on the lending data of New Zealand banks, which served to differentiate the credit risk of borrowers, etc. Fatemeh (2016) combined genetic algorithm and clustering algorithm with the traditional credit scorecard system, and after empirical analysis found that the credit scorecard system improved by data mining algorithm could better complete the assessment of bank loan quality. Masoumeh(2017) introduced the integrated learning algorithm into the field of credit risk assessment of commercial banks and found that the bagging integrated algorithm was more effective in assessing credit risk than the traditional data mining algorithm after empirical analysis. The model is also efficient and practical. Yanhong constructed a random forest model on P2P lending data in Germany, and then compared the assessment results under the random forest model with the traditional method, and found that the random forest model is more capable of assessing the credit risk of borrowers.

Data mining theoretical model

Data mining refers to the process of extracting useful information from the data of the issue under study through computer software or platforms when dealing with certain real-world issues and providing technical support for solving real-world issues. The emergence of data mining techniques is closely related to other theories, and the relevant theories of statistics and artificial intelligence are reflected in data mining techniques. By the nature of data mining itself, data mining models can be divided into single data mining models and integrated class data mining models. Single data mining models include logistic regression models, neural network models, decision tree models, SVM models, etc. Integrated class data mining models include random forest models, XGBoost models, bagging model combinations, Boosting model Combination, Stacking model fusion, etc. Next, we introduce the algorithmic principles of several commonly used data mining models.

Logistic regression model

The logistic regression model is a classical classification model used to deal with Bernoulli distribution data issues. The basic formulas of the logistic regression model for the binary classification issue are shown in Equation 1.1 and Equation 1.2.

$$P(Y=1 \mid X) = \frac{1}{1+e^{-\left(\alpha_0+\Sigma_{i=1}^{n}\alpha_i x_i\right)}}$$

$$P(Y=0 \mid X) = \frac{e^{-\left(\alpha_0+\Sigma_{i=1}^{n}\alpha_1 x_i\right)}}{1+e^{-\left(\alpha_0+\Sigma_{i=1}^{n}\alpha_1 x_i\right)}}$$

Where the dependent variable Y may belong to two categories 0 or 1, n is the number of independent variables, and the output is the probability that Y belongs to category 1. The probability value P can only be between 0 and 1. When P is closer to 1, it means that the more likely Y belongs to category 1, otherwise, the more likely Y belongs to category 0.
Among the data mining models, the algorithm principle and model structure of the logistic regression model is relatively simple. Compared with other data mining models, the characteristics of the logistic regression model are more obvious. First, the output form of the logistic regression model is easy to understand, and the coefficient value Q of the output represents the effect of the independent variable on the target variable, and the output result P represents the possibility of the data being classified into this item; second, compared with more

complex machine learning models such as support vector machines and random forests, the internal structure of the logistic regression model is simple. Third, the logistic regression model can be applied to both categorical and numerical independent variables without any additional processing, which makes it more widely used.
Of course, logistic regression models also have some significant drawbacks, such as the generalization ability of model training may be reduced or even lost under the influence of multicollinearity. Logistic regression is also not good at dealing with the differentiation of nonlinear variations on probability values. The treatment for the above issues can be solved by variable dimensionality reduction, and in addition, regularization can be used to adjust the coefficients of each variable to reduce the influence of multicollinearity.

Support vector machine model

Support vector machine model (SVM) is a frequently used machine learning model, and its algorithm is based on the principle of statistics, which aims to minimize the empirical risk and confidence range of the model by minimizing the structured risk. The SVM model is essentially a classifier that defines the maximum distance in the feature space, and it can differentiate data in terms of type by constructing a "hyperplane" that lies between the points of different categories. The SVM model is essentially a classifier that defines the maximum distance on the feature space by constructing a "hyperplane" between different categories of points, thus enabling the differentiation of data in terms of type. Therefore, the training of the SVM model is based on the principle of distance maximization. Under the condition of nonlinearity and a small number of samples, SVM model prediction is better than other data mining models.
SVM models are usually classified into linear SVM models and nonlinear SVM models according to the nature of the classification issue to which they are applied. Linear SVM models are mainly applicable when the data set is linear. For linear SVM models, the mathematical expression of the model and the classification decision function are shown below 1.3.

$$\omega^* \cdot \mathrm{x} + \mathrm{b}^* = 0$$
$$\mathrm{f(x)} = \mathrm{sign}(\omega^* \cdot \mathrm{x} + \mathrm{b}^*)$$

The training goal of the linear SVM model is to maximize the majority of sample data to be correctly classified. However, in real life, many classification issues are nonlinear, when the training effect of the linear SVM model cannot reach the classification requirement, it is necessary to choose a nonlinear SVM model to solve the issue. The nonlinear SVM model mainly uses the kernel function k(×, Z) nonlinear transformation to transform the nonlinear issue into a linear one. The kernel functions available under general conditions include polynomial kernel function, string kernel function, Gaussian kernel function, etc. Through the kernel functions, the correspondence between the original space of the model and the new space is created, and the research issue is converted from a nonlinear issue to a linear one.
Compared with other data mining models, the structure of the support vector machine model is more complex, and the SVM model has other features that distinguish it from other models, including the fact that the SVM model does not perform model construction and model prediction through the traditional statistical analysis process but through the structural reasoning from the data training set to the data testing set, which can clarify the issue-solving ideas. In addition, the principle of model training through SVM is more evident in the role of a few

branch vectors, which helps researchers to extract the most influential data in the dataset for model prediction. However, the training efficiency and prediction efficiency of SVM models are affected by the sample size, and when the sample size increases significantly, their model training efficiency and prediction efficiency will also decrease significantly.

Decision tree type models

Decision tree model

The decision tree model is a data mining model with a simple algorithm idea, which is more suitable for nonlinear issues. The essence of the decision tree model is a classification function approximation method. That is, based on the known attributes in the training set and the results obtained after analysis, a tree of data classification rules is trained, and the tree classification rules are applied to the model prediction of the test set.

The decision tree model has three types of data nodes, namely, root, middle, and leaf nodes. Each of these three types of data nodes represents an attribute of the sample, and there can be multiple paths from each node as the starting point, and several paths represent how many attributes there are. The specific model classification method is root node to some path to some intermediate node and finally to some leaf node, and the path from the root node to leaf node can be regarded as a certain data classification rule. The model form of the decision tree is the set of all data classification rules. For any data, there is one and only one path in this decision tree, which determines the uniqueness of the model output for that data. With this set of paths, the prediction set also completes the model prediction. According to the algorithm principle of the decision tree model, the model application process is divided into three steps: first, the training set is applied to the model training to generate the inverted tree topology; second, the training set is used to generate all the paths from the root node to the leaf node for the tree structure generated in the first step, and the set of these paths is the data classification rules; third, the test set is applied to the classification rules in the second step to complete the model prediction.

Compared with other data mining models, the decision tree model also has some remarkable features. Firstly, the decision tree model has a better explanation in business; secondly, the decision tree model has better applicability to data, whether it is discrete data or continuous data, the decision tree model has a good processing effect; finally, the decision tree model has good processing effect whether it is a small number and training set or a large number of training sets.

First, under the influence of the overfitting effect, the model training results of the decision tree model will be distorted; second, the decision tree model is not applicable to high-dimensional variable data, and the effect of model training and model prediction will be degraded to a large extent under the high-dimensional data; third, the stability of the decision tree model will be significantly reduced in the face of the interference of outliers in the data. Therefore, decision tree models with integrated learning ideas (such as random forest models and XGBoost models) are born.

Random forest model

Random forest model is a data mining model with integrated learning ideas, which is essentially a classifier model with many decision trees. The random forest model applies the integrated learning idea of bagging. In the model training, a sample is voted on the output of all the decision trees in the random forest model, and the voting result is the model classification result of that sample. By performing the above operation on all samples, the model training is completed and the specific structure of the random forest model is obtained.
Compared with the single decision tree model, the random forest model has many advantages: first, the random forest model has better model training and model prediction when facing the sample data with outliers; second, the random forest model has better model training and model prediction when facing the data with high-dimensional variables; third, the model construction of the random forest model is completely data-driven, so that no prior knowledge is required in building the model, which greatly extends the application scenario of the random forest; finally, the random forest model can also have a better solution in dealing with the overfitting issue.

XGBoost model

Like the random forest model, the XGBoost model also applies the idea of integrated learning, and the model structure is also a classifier model with many decision trees. The output of the XGBoost model is the result of accumulating all the decision trees in the model. The researcher also applies including loss function, regularization, and parallelization algorithm design to the model structure design of the XGBoost model. Compared with other data mining models, the quadratic convergence feature and multi-path processing feature of the XGBoost model make the model training and model prediction significantly more effective, and the XGBoost model is also more suitable for processing structured data. In terms of the practical application effects of the model, although the XGBoost model has not been applied in various industries, the XGBoost model has shown good model prediction effects in business modeling competitions.

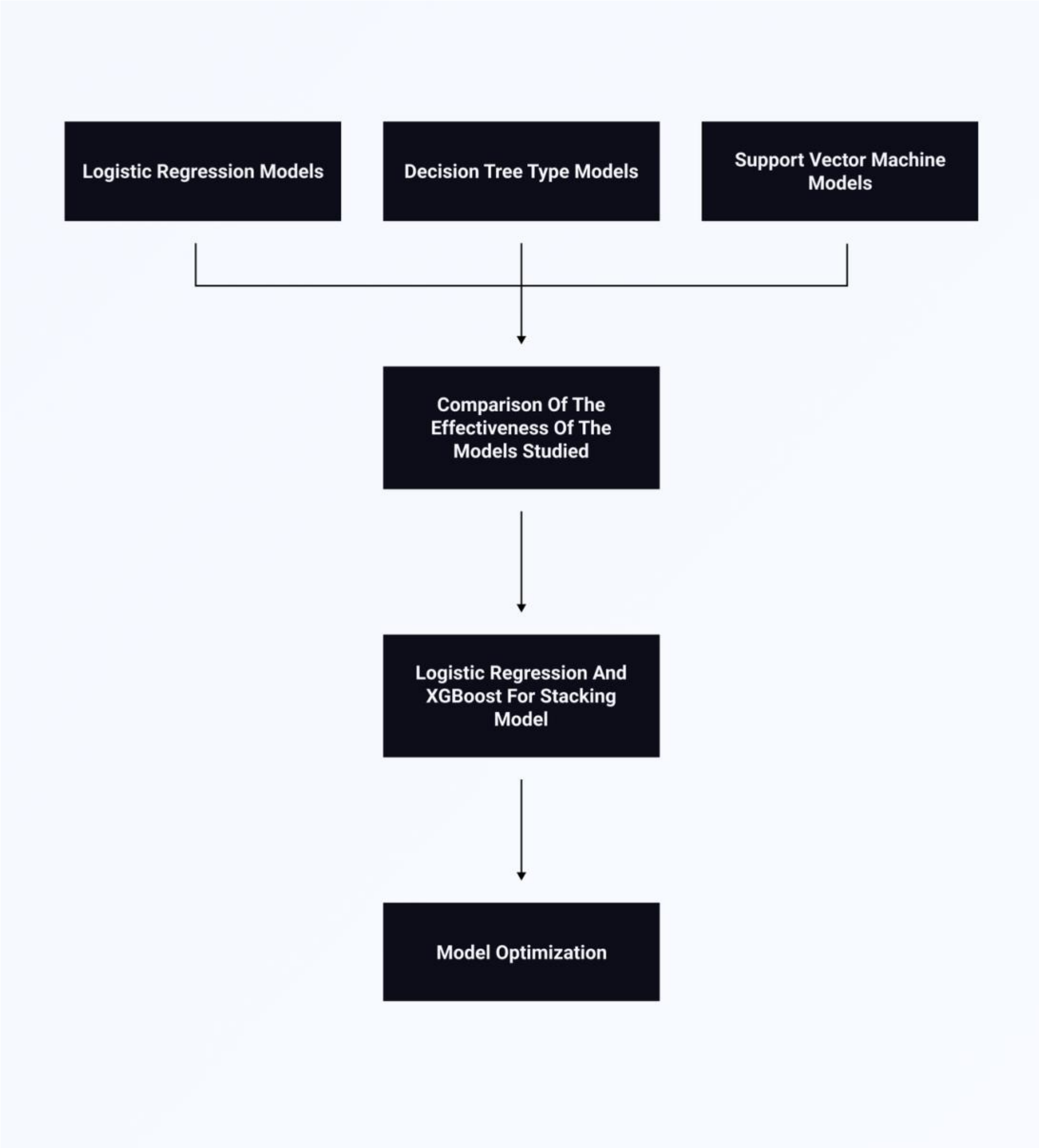

Figure.1

Design of credit scoring model based on blockchain data

Method selection for credit scoring model optimization

The optimization method of the on-chain credit scoring model based on data mining technology is mainly divided into two stages. The first stage is to select a single model with better model prediction ability and model efficiency through model comparison of a single data mining model, and the second stage is to use the idea of integrated learning to carry out Stacking model fusion of the single model selected in the first stage and compare the effect of the model after Stacking model fusion and the effect of the single model to complete the optimization of the on-chain credit scoring model.

Sub-model selection method

For the evaluation of the single model, the two main aspects are analyzed from the model's modeling efficiency and the model's prediction effect. Different data mining models perform differently in these two evaluation metrics; for models with simple algorithm structure, the former performance is generally stronger than the latter; while for models with complex algorithm structure, the latter performance is generally stronger than the former.
The models to be compared include simple logistic regression models and decision tree models, as well as more complex SVM models, etc. In addition, the more popular data mining models are also involved in the comparison, such as random forest models, XGBoost models, etc. After modeling and predicting a single data mining algorithm, the term evaluation effect of the test set is compared, and the single data mining algorithm with a better test effect is selected as a sub-model into the model fusion.

Stacking model fusion

Using a single data mining algorithm to build a model based on the actual issue, the model built by the single data mining algorithm has achieved remarkable results in model evaluation. However, the issues that arise are equally difficult, such as the model is not accurate enough and there are limitations in the amount of data processing. Therefore, the idea of integrated learning is used to optimize the single data mining model in order to solve the above issues. This paper selects the Stacking model fusion method in the integration learning according to the actual issue of credit score in the evaluation chain. Unlike the Bagging algorithm and Boosting algorithm, the Stacking model fusion approach is widely used in recent years. Stacking model fusion is considered one of the optimal integration learning algorithms.

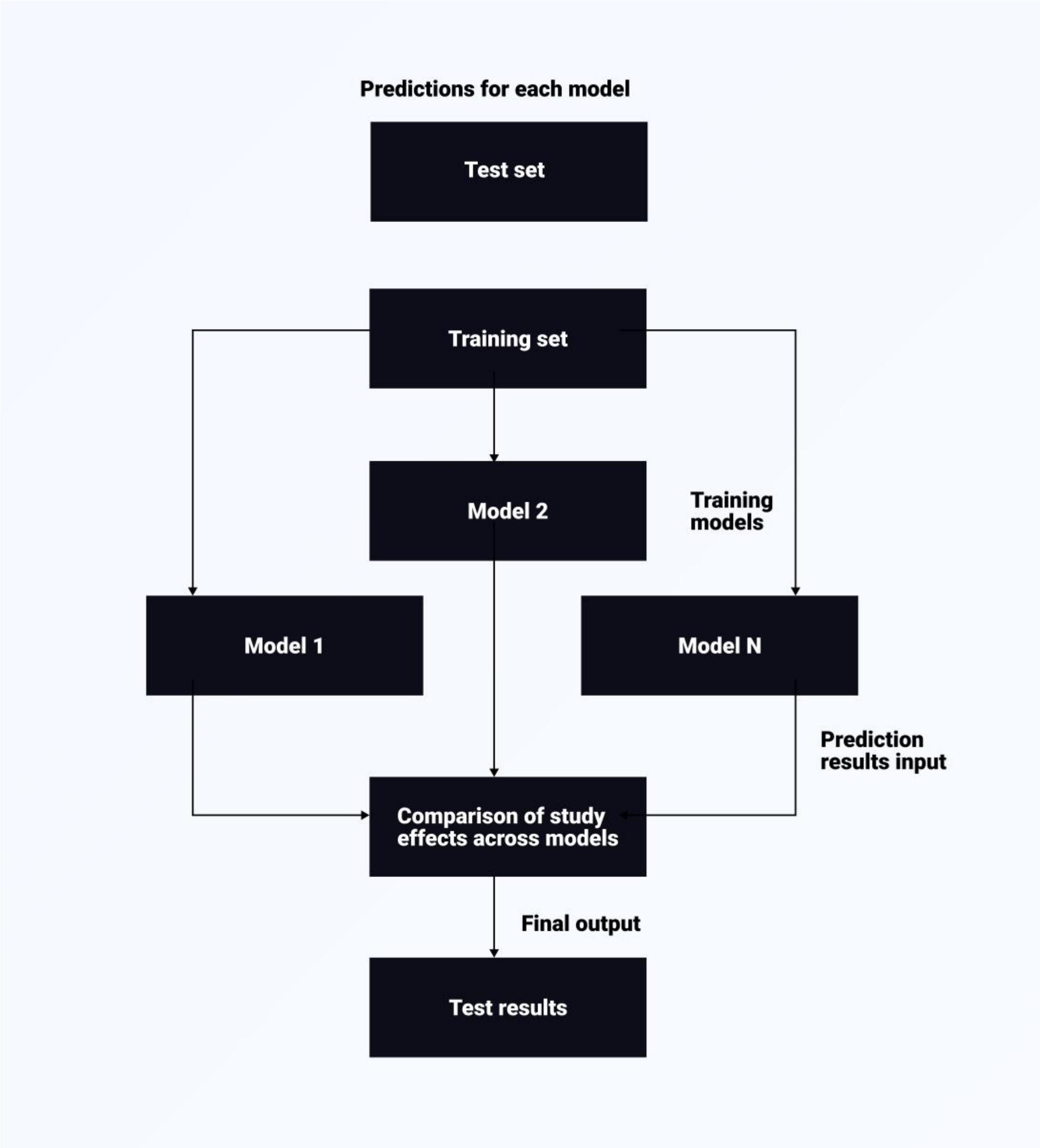

Figure.2

The Stacking model fusion algorithm combines the advantages of various types of sub-models in the first stage by its unique algorithmic characteristics, which makes the prediction results of Stacking model fusion better than any single model. Due to the nature of Stacking model fusion itself, Stacking can minimize the degradation of model evaluation metrics due to model overfitting, which means that Stacking model fusion can maintain high model prediction metrics despite large data volumes and high dimensional variables.

Smart contract-based credit score sharing model

The smart contract-based data sharing interaction model removes the participation of third-party web servers, solves the data trust issue of centralized management, and ensures that both data suppliers and demanders can interact flexibly and reliably through transparent bookkeeping. To facilitate data management, the data providers are divided into two roles, data source and data owner, and Figure 3 shows the specific role division.

Character	Description
Data source	Refers to a database management system that has the ability to manage and store data sets and provide access and even download channels.
Data owner	Refers to blockchain DApps user that has ownership of the dataset.
Data Requester	DApps, researchers, etc., who need access to datasets.

Table.1

The blockchain-based data sharing interaction information is described as follows, and the smart contract-based data sharing interaction model is shown in Figure 4 below.

(1) The original information in the data source is processed to obtain the relevant information RT, which consists of the shared data keywords, the access path DAP of the complete data (which can be a URL, URI or other access paths), and the public key address of the data owner.

(2) The data owner publishes RT, which is stored in the form of Data on the blockchain, while the relevant information ST for retrieval is stored in a distributed hash table. st consists of data keywords, the hash value of the data access path, and the public key address of the data owner.

(3) The data requester retrieves the desired shared data information through the table and gets the public key address of the data owner;

(4) The data requester sends a request QT to the blockchain network based on the public key address of the data owner, which consists of the hash value of the shared data and the address of the data owner.

(5) The data requester's identity is verified by AuthoList to get the stored information Data on the block.

(6) The data requester accesses the data source through the path information in the decrypted Data to complete the data sharing interaction.

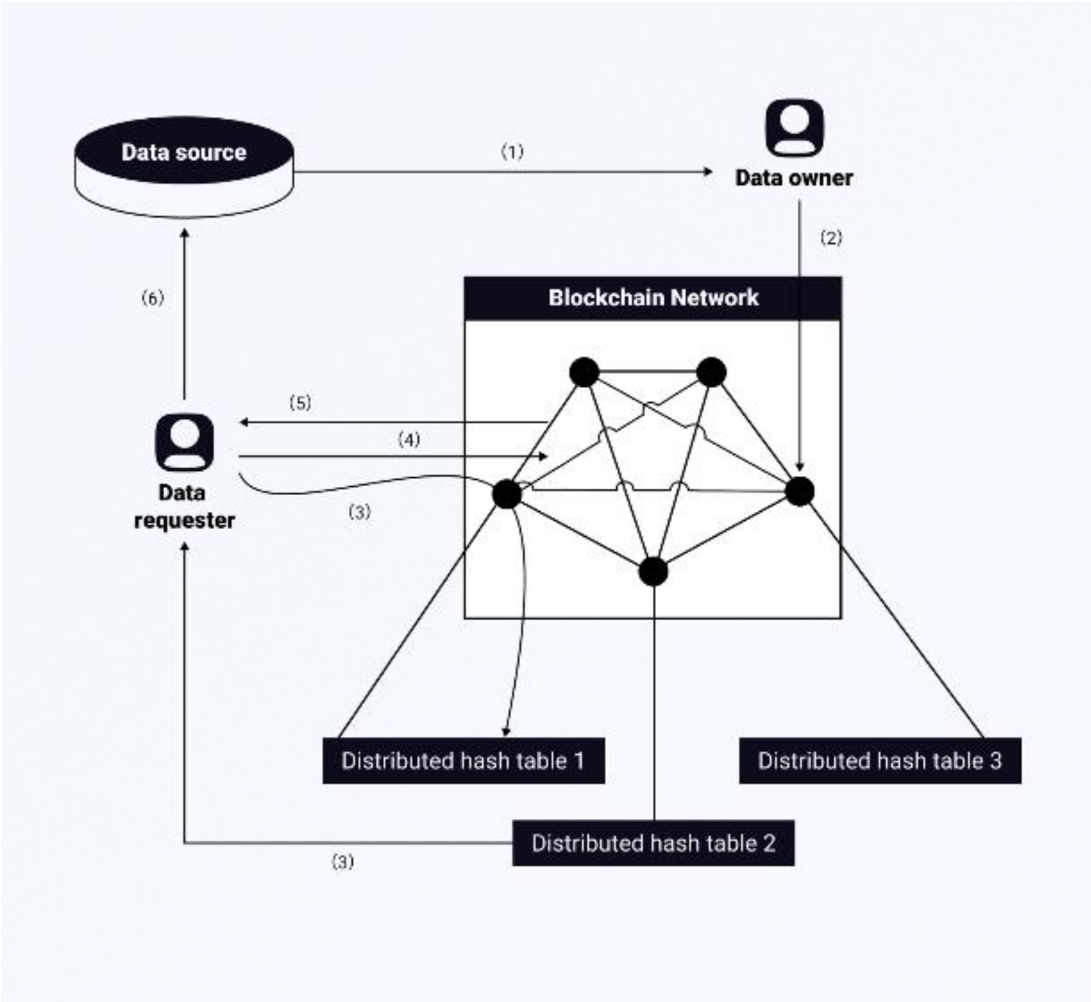

Figure.3

To facilitate a detailed and standardized description of the interaction process, the relevant data symbols are defined in Table 5.

RT	The relevant information is obtained after processing from the original information.
QT	Request information for data requesters.
name	Shared data keywords.
DAP	Access path to complete data.
owner	Public key address of the data owner.
H {}	Perform hashing algorithm processing.
sign {M， X}	Sign the message M with X's private key to indicate that the message was sent by X.
{M} X	Encrypt message M with X's public key and send this message to X.
{M} -X	Decrypt the message M using X's private key.
table {}	Retrieve database tables table
AuthoList {X}	Authenticate against X

ski	i's private key.
pki	i's public key.
O	Data owners.
D	Data source.
R	Data requesters.
Xi → Xj: M	i transmits a message M to j.
X: S {}	X Perform the processing of S {}
M= {M1|M2|...Mn}	M consists of M1, M2.... One or more of Mn

Table.2

Data sharing interactions protected by blockchain asymmetric cryptography are modeled in conjunction with the terms in the table, as detailed in Table 5.

Data sharing interaction modeling table

// raw data from the data source is processed
(1) D → O : {RT }
RT → {name, DAP, owner}
DAP →{URL | URI | FTP | OtherDataL ocation}
// Data owner releases data
(2)O → * : Data →sign{{name, hash{DAP}, owner , timestep}, sko}
//Data requester retrieves data
(3) R : table{name, hash{DAP}, owner}
//Data requester sends data request
(4) R→* : QT {hash{DAP}, owner}
//Data requester gets the returned data
(5) * → R : {AuthoList{R}, name, {hash{DAP}} pkR}
// The data requester decrypts the corresponding data and accesses the data source
(6) R : {hash{DAP}} → skR

Design of smart contract-based data sharing mechanism

This section provides the general design of the data-sharing system based on smart contracts, which includes the local database, smart contracts, and front-end pages. According to the requirements of the smart contract-based data sharing application, the design of functional modules was carried out, including the storage function of shared data, the smart contract consensus function, the retrieval function of shared data, the access request of shared data, and the service module function. The application development environment required for the experiment was established, and the implementation of the two main functions, the storage function of shared data and the request and access function of shared data, was completed.

General system design

The data-sharing system based on cloud services is based on a third-party web server for data storage and sharing. This paper removes the third-party web server and designs a functional implementation architecture based on smart contracts, decentralized and trusted data sharing with smart contracts as the core. The overall design architecture involves four parts: the local data storage device, the smart contract network built by smart contract nodes, the smart contract, and the front-end of the application, and the total design is shown in Figure 4 below. Each part is described in detail below.

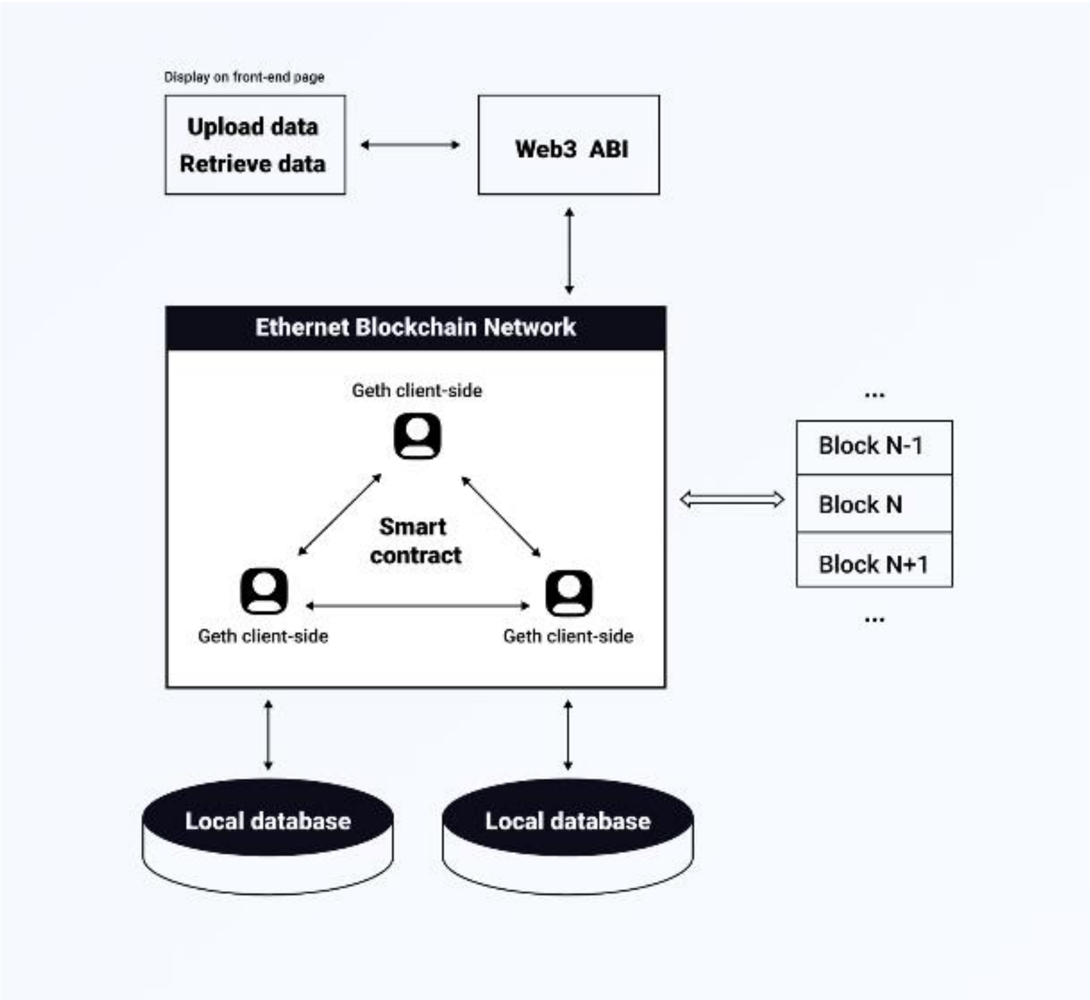

Figure.4

(1) Local database

This layer is responsible for the storage of shared data. The shared data of the data owner is stored locally, and during the sharing process, no one has ownership of the shared data except the data owner. The shared data in the local database is not in the smart contract network, but after data processing through metadata, the authenticity data identifier is finally stored in the smart contract, and it is this identifier that is transmitted in the network.

(2) Smart Contracts

This layer is based on the business logic that needs to be implemented in the data-sharing scheme of the smart contract, and the smart contract scripts for shared data storage and shared data requests and services are written flexibly using a suitable programming language. The smart contract script is strictly implemented by all nodes of the network.

(3) Blockchain
This layer is the core part where nodes participate in the smart contract network through Geth clients, responsible for block generation and maintaining the operation of smart contracts. The storage of shared data is done before the data is shared, and additionally provides the service of shared data request and response. Each node on the smart contract can have two roles at the same time, i.e., data owner and data requester.
(4) Front-end page display part This layer enables each participant to realize the storage of shared data and shared data request and service based on the front-end page, and finally complete the interaction of data sharing, provided that the smart contract network is running normally and the contract is deployed successfully.

The functional design of the system

According to the application requirements of smart contract-based data sharing, the decentralized data sharing system based on smart contracts mainly involves five functions: local data storage function, shared data storage function, shared data retrieval function, block consensus function, and data sharing request and service function.
Among them, the storage function of shared data, the retrieval function of shared data, and the request and service function of data sharing are the three main functions. In the implementation diagram of the system function design, the local data storage function refers to the local storage of the raw data of the shared data, which can be any type of data, provided that it can be processed with metadata. To facilitate the later experiments, different types of data are selected for local storage or directly using data from the local server. The functional design of the smart contract entity is specified below.

Shared data storage module design

The shared data storage function is a hash of any type of locally stored data after metadata processing to obtain a hash value. This hash value is the authenticity mark of the metadata after the digital signature of the data owner, and finally, both this hash value and the authenticity mark are stored on the smart contract. This method is used to create the authenticity identifier of the data. The data structure that needs to store the shared data, i.e., Data structure, is provided to the smart contract node, including metadata information such as keywords of the shared data, the hash value of the complete shared data, timestamp, digital signature, and data owner. The smart contract node device is responsible for collecting this information. The corresponding script file is executed to verify whether the storage function is valid and eventually successfully stored into the smart contract. This function is similar to the transaction function in Ethernet smart contracts.

Smart contract consensus function module design

The smart contract consensus function involves the core content of smart contracts. It provides the function of having a consistent ledger of nodes in the whole network, and is an important implementation for the packaging of transactions, the construction of blocks, and the composition of blocks into a chain. All shared data storage records must be verified by this function before forming a block, which is then verified by the nodes in the network and the block is verified by the consensus protocol before being linked to the tail of the smart contract. The consensus protocol defines the rules of the smart contract, such as the agreement of the smart contract nodes, the consistency of data, etc., and the order in which the nodes reach global transactions. This function is similar to the proof-of-work mechanism of the Ethereum smart contracts.

Shared data retrieval function module design

The shared data retrieval function refers to the data requester's search in the database, using the database's primary key search method to find the corresponding public key address by retrieving keywords or using the database's fuzzy query method to find the corresponding shared data content for subsequent data requests and services. The retrieved database stores three types of information, one is the public key address of the data owner, another is the keyword of the shared information, and the last is the hash value of the local shared data, which is only used for the provision of experimental trusted data sharing. Since the number of records of shared data in the experiment is not very large, the manual input method is chosen to record the number of records of shared data into the database.

Access request and service module design for shared data

The access request and service function of the shared data provides the function of data visitor to request access to the shared data, and the public key address of the corresponding data owner is obtained by retrieving the keywords of the shared data. shared data information. The data owner's shared data information stored in the block is returned, in which the link to the complete data is encrypted with the requester's public key by the asymmetric encryption algorithm and output. When the requester gets the information about the block, he decrypts the encrypted complete data link portal with his private key. This function is also similar to the transaction function of Ethernet smart contracts.

Functional implementation of the system

Based on the data-sharing interaction model in the previous section, this paper mainly uses the Ethernet smart contract as the smart contract deployed by the smart contract, and each function is also realized through this smart contract, mainly realizing three functions of data sharing based on the smart contract, which are: the shared data storage function, the shared data retrieval function and the shared data request and service function. The functional structure of the program is shown in Figure 5. In the whole project, the app directory is the default directory, which is used to store the front-end files, usually including JS and CSS files; the contracts directory is used to store the back-end files, where the smart contract files containing business logic are stored; the migrations directory is used to store the published script files, truffle.js is the truffle configuration file, test directory is used for testing smart contracts and applications, package.json contains the installation package needed to run the system.

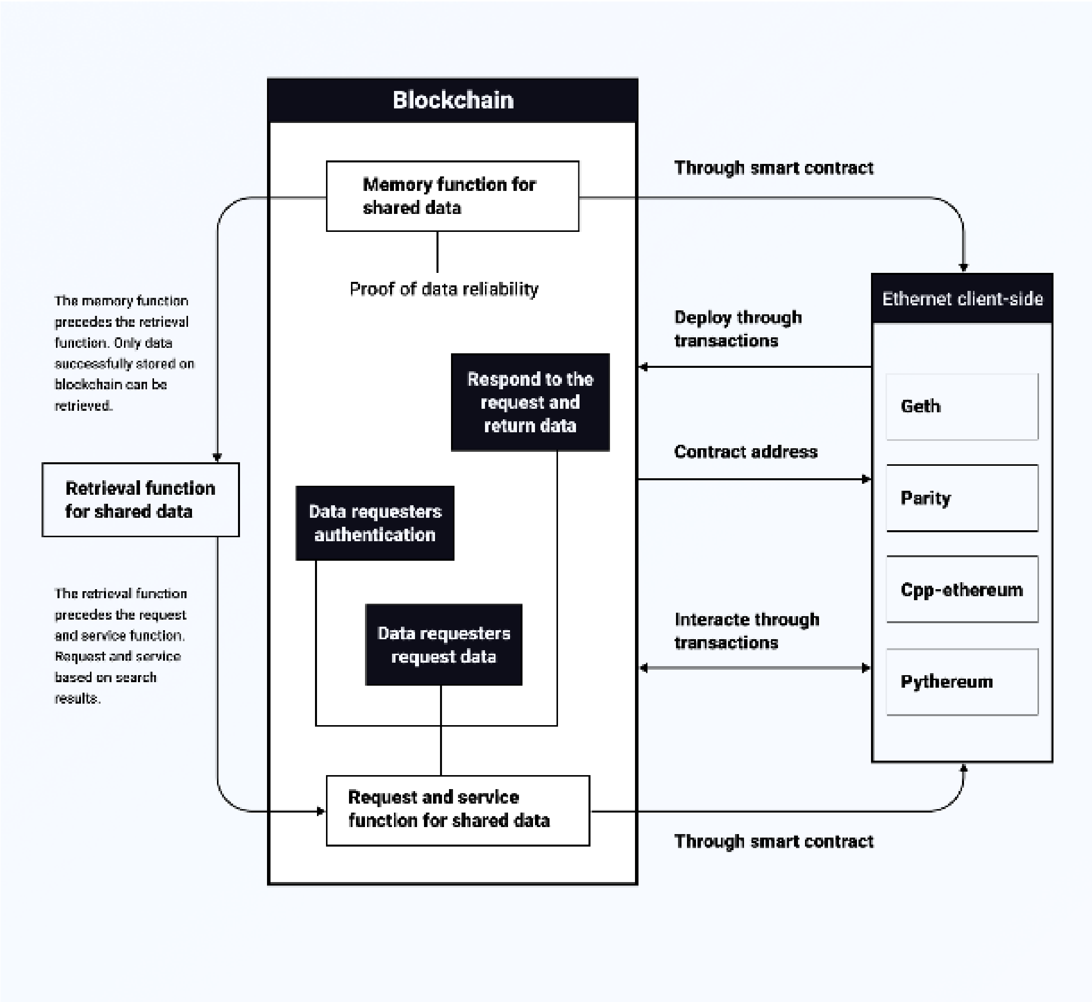

Figure.5

First, start an Ethernet node, Geth, and write a smart contract file using the Solidity programming language with the .sol suffix. compile the smart contract file, the smart contract is converted into a bytecode that the Ethernet virtual machine can recognize, and obtains a binary interface ABI for the account to interact with the smart contract. when creating a contract, the account uses the bytecode When the contract is created, the account will broadcast the bytecode as the passing parameter of the transaction to the whole network for verification, and the contract will be created successfully after valid verification, forming the transaction record on the smart contract. When the contract is called, the execution of the smart contract also exists in the form of a transaction. The account gets the result of the contract's operation through the ABI interface and records the process as a transaction stored in the smart contract. Whether compiling or deploying a smart contract, a certain amount of fuel is consumed, and the contract initiator is required to sign the contract with his private key. After passing the proof-of-work verification, the contract code is successfully stored on the Ethernet smart contract. The information contained in the smart contract used for data sharing is shown below.

DataSharing
+ struct Data{
bytes name;

```
bytes hash;
 bytes dsign;
uint256 timestamp;
address owner;
}
+ store(bytes32 name, bytes hash, uint256 timestamp, bytes dsign) public returns (bytes) +
query(address owner) public returns
(bytes32, bytes32, uint256, address, bytes)
```

The above smart contract contains the following properties and functions.
name, the keyword selected from the locally stored shared data and stored on the smart contract.
hash, the hash value of the shared data link entry.
dsign, a digital signature based on a decentralized timestamp.
timestamp, the local time when the data was stored to the smart contract, i.e. the timestamp.
owner, the public key address of the data owner.
store() function performs the shared data storage function. Each time the shared data is stored, the global state of the smart contract is updated and considered as one transaction.
query() function performs the request and service function of the shared data, making a request to the smart contract network for access to specific data, and responding based on the identity of the visitor or not. This function also updates the global state of the smart contract and executes a transaction.

Shared data storage function implementation

Before the storage function is implemented, the storage of raw data is completed and metadata processing is performed to store the relevant data representing metadata information into the smart contract. The experiments use a special way of storing data that combines two methods of storing data: using accounts and using events. Storing data using events means that each interaction with the smart contract is recorded as a transaction on the smart contract. Both of these are implemented in the smart contract. The type of data stored is usually a value type or a reference type. The non-tamper-evident nature of smart contracts allows an attacker to tamper with the data in a smart contract only by redeploying the smart contract to make the changes, which are also recorded on the smart contract. Smart contract technology can securely record transactions between two parties without the need for a trusted third-party institution while providing a way to protect privacy using a variable public key as an identity, while smart contract security is ensured by mining new blocks through proof-of-work to solve cryptographic challenges.

The design of the specific smart contract contains a store storage function that will have a data identifier representing metadata passed to the store function, the data identifier is passed as a parameter to this function and stored in the smart contract, the data identifier contains the metadata information of the shared data, including which name, hash, dsign, timestamp, owner, respectively represent the subject information keyword of the shared data, the hash value of the link entry of the complete data, digital signature, timestamp, and data owner. The code is shown below.

```
contract DataSharing { function store(bytes32 name, bytes hash, uint256 timestamp, bytes dsign)
public returns
(bytes) {
if(msg.sender != owner) throw; address owner = msg.sender; bytes32 message =
prefixed(keccak256(name, hash, timestamp)); address recovered = recoverSigner(message,
dsign); require(recovered == msg.sender); Datas.push(Data({
name: name,
hash: hash,
timestamp: timestamp,
owner: owner,
dsign: dsign })); Dsign(name, hash, timestamp,owner, dsign); return dsign;
}
}
```

A key point of data trustworthy storage is digital signature technology. In this paper, we use keccak256 algorithm for data processing of metadata, and the proposed Data structure is the basic structure representing data authenticity. In the process of digital signature equation, h() represents the hash function to generate a hash value, m represents the message information, timestamp represents the timestamp, private represents the user's private key, and sign{} represents the digital signature scheme. The process of digital signature generates a unique authenticity identifier dsign. through the store storage function, the shared content keyword name, digital signature dsign, timestamp timestamp, hash value hash of the link entry of the shared data, and public key address owner of the data owner are permanently stored in the ethereum smart contract.

$$dsign=sign\{h(h(m),timestamp),private\}$$

The above code in the prefix(keccak256(name, hash, timestamp)) and the local data hash sig is exactly the application of the above formula. In addition where the recoverSigner function is used to implement the verification of the digital signature. As shown in the table below.

function recoverSigner(bytes32 message, bytes sig) public returns (address)
{
uint8 a; bytes32 b; bytes32 c; (a, b, c) = splitsig(sig); return ecrecover(message, a, b, c); }
	function splitsig(bytes sig) public returns (uint8, bytes32, bytes32) { uint8 a; bytes32 b; bytes32 c; assembly { c:= mload(add(sig, 32)) b := mload(add(sig, 64)) a := and(mload(add(sig, 65)), 255) }
	
	if (a < 27) a = a + 27;
	
	return (a, b, c);
	}

Table.3

The ecrecover() function in the Ethernet platform is a function for digital signature verification, and the return value is the public key address of the signer. The r, s, and v in the function are the signature string after intercepting the signature result, and the interception formula is r = signature[0 : 64], s = signature[64 : 128], and v = signature[128 : 130]. When the returned public key address is the public key address of the real signer, then the digital signature is correct.

Shared Data Access Request and Service Function Implementation

The smart contract contains a shared data access request and service phase, which is divided into two parts: one is to retrieve and find the block where the target data is located, and the premise of this thesis is to find the public key address of the data owner of the required shared information based on the subject information keywords of the shared data. contract sends data access requests and services. After the shared data is stored on the smart contract, the data visitor makes a data access request to the smart contract based on the public key address of the corresponding data owner obtained by the data visitor after information retrieval. In the access request, the data owner first verifies the identity of the visitor, i.e., checks whether it is authorized in the AuthoList, and if it is authorized, returns the corresponding shared data information stored in the block, in which the link entry of the complete data is encrypted with the visitor's public key through the asymmetric encryption algorithm and the output. When the visitor gets the information about the block, he/she decrypts the encrypted complete data link entry with his/her private key. Otherwise, it has to be authorized first, which is authorized by the verification and approval of the nodes in the whole network. The specific implementation of the smart contract contains a query request function that runs as shown in the figure below.

```
function query(address owner) public returns(bytes32, bytes32, uint256, address, bytes){
bool autho = AuthoList[msg.sender];
if(autho == true){ uint len = Datas.length;
for (uint i = 0; i <= len - 1; i++) {
bytes memory sign = Datas[i].dsign; if (isEqual(sign, dsign)) { return (Datas[i].name,
recover(Datas[i].hash), Datas[i].owner, Datas[i].dsign); } }
Datas[i].timestamp,
}else{ addAuthorize(msg.sender);
}
}
```

The addAuthorize function verifies the identity of the visitor and checks whether he/she is a node in the smart contract network, and only needs to verify whether he/she has a pair of public and private keys to prevent nodes not joined to the smart contract network from participating in data sharing. In addition, the recover function is to return the corresponding shared data information stored in the block after the visitor's identity verification is passed, where the link entry of the complete data is encrypted with the visitor's public key through the asymmetric encryption algorithm and output.

This chapter specifies the functions that need to be implemented in the smart contract-based data sharing scheme, including the shared data storage function, the smart contract consensus function, the shared data retrieval function, the shared data access request and the service module function. The main functions of the smart contract-based data sharing scheme are implemented, which are the storage function of the shared data, the request and access function of the shared data. For the storage of shared data, the public key is used to identify the user identity, the complete data is stored locally, the access portal of the data is stored on the smart contract in the form of hash value, and the private key is used to store the signature of the transaction; for the request and access of the shared data, the prior authentication of the authorization status of the data requestor is realized. Both generate transactions and are recorded on the smart contract.

Test Implementation

In the smart contract-based decentralized data sharing experimental system, the storage function unit of shared data and the request and service function unit of shared data are used as black boxes, respectively. Test points are determined according to the application requirements of decentralized and trusted data sharing. In the storage function unit of shared data test unit, the trusted representation of data is the test point; in the request and service function unit of shared data, the trusted interaction between two parties of data sharing is the test point. The following figure shows the functional diagram of the system using the functional diagram method.

Overview of credit score usage scenarios

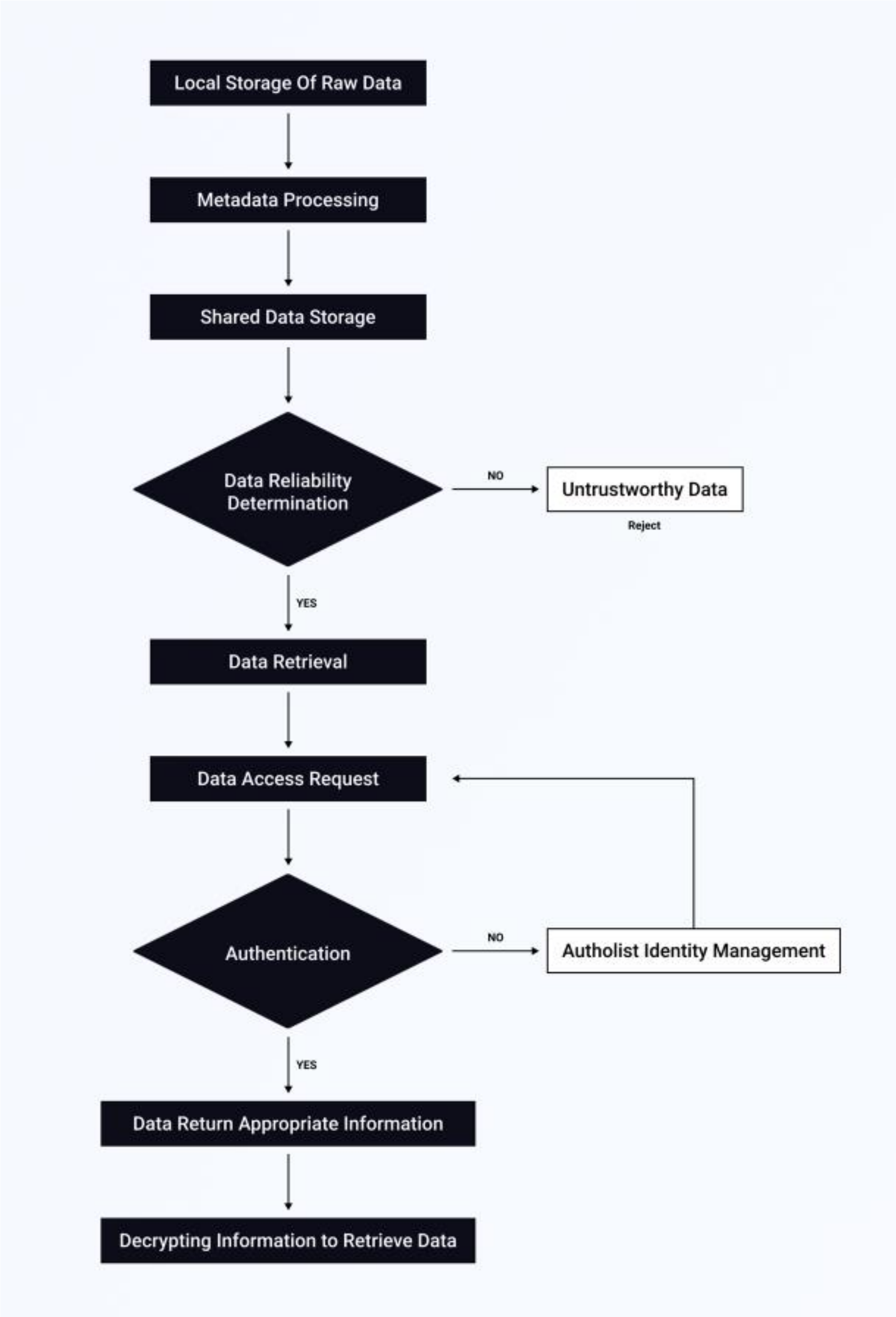

Figure.6

Overview of credit score usage scenarios

This section states the credit score usage scenarios.

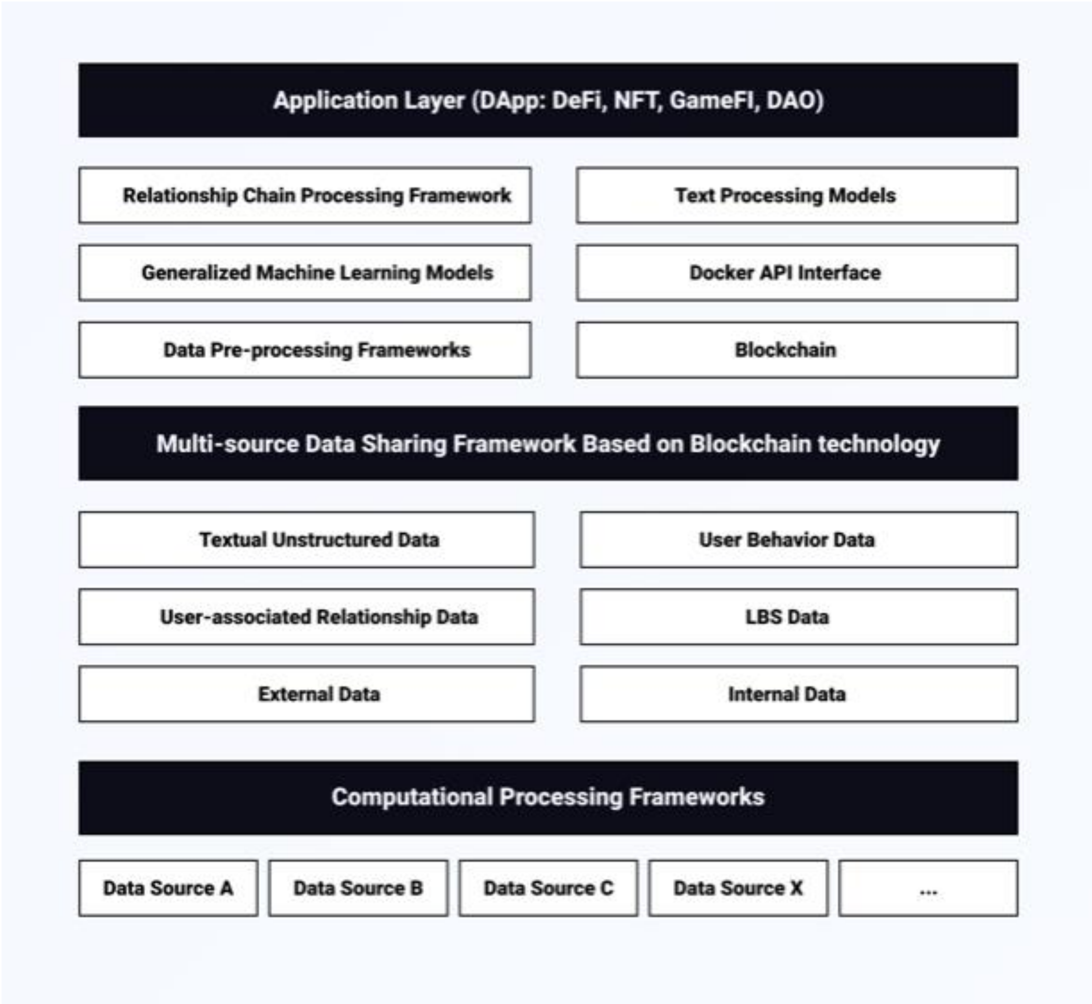

Figure. 7

The Symbol of Honor

MetaVisa Protocol issues a badge of Honor by evaluating the credit rating of an address. Each badge will be an exclusive NFT owned by the address. The badge will be an important symbol for the user in the Metaverse. Users can also display the badges through traditional social platforms such as Twitter, Facebook, LinkedIn, and Instagram.

Login Credentials

MetaVisa Protocol credit system is associated with the public key, making the MetaVisa Protocol credit system highly identifiable. Based on the W3C WebAuthn, the MetaVisa Protocol credit system can be used as a credential to log in to third-party applications in Metaverse. The login method will be changed from entering the account password to using the private key associated with the MetaVisa Protocol credit system to sign the login operation, greatly improving security and user experience.

Integrate with Game

Games in Metaverse can be combined with the MetaVisa Protocol credit system to achieve special rewards distribution, game asset credit transactions, etc. In addition, when game players form a guild in a game, they can effectively manage the guild members according to the different MetaVisa Protocol credit levels.

Integrate with DeFi

The decentralized lending platform can combine with the MetaVisa Protocol credit system to improve product experience, such as adding credit lending services, etc. In addition, the IDO platform can also have different credit levels and carry out more effective and reasonable quota allocation.

Integrate with DAO

Most DAO governance is determined by the number of tokens held to determine the voting weight, but this method can easily lead to the issue of centralized domination by a small number of users. DAO can be combined with the MetaVisa Protocol credit system, such as addresses with higher credit scores will have more voting weights.

User growth

In order to attract users in the early stage, Devs usually use airdrops for marketing, but airdrop hunters with thousands of addresses can easily ruin the development of early projects. So instead, devs can use the MetaVisa Protocol credit system to screen out trusted users, thereby improving marketing efficacy quickly.

Customer Management

The devs or the community can formulate a corresponding membership system or point redemption system based on the MetaVisa Protocol credit system to motivate users, maintain user relationships, and thereby enhance user loyalty.

Privacy and Information Security

- MetaVisa Protocol will not invade your privacy and collect personal data other than the address behavioral acts on the chain.
- MetaVisa Protocol has formulated a special "Credit Data Security and Privacy Protection Policy," dedicated to protecting user privacy.

- To ensure the security of your private information, when we output information externally with your authorization, in principle, we will not output your original detailed information but information that has been processed by desensitization and obfuscation.
- Established a complete data security management and control process and system and adopted technical means to ensure that the corresponding systems and processes are fully implemented.
- We review the data security capabilities of all cooperative organizations. Only those organizations that meet the corresponding standards will cooperate. At the same time, we will export information security protection standards to these organizations to enhance the information security protection capabilities of the cooperative ecosystem.

Economic Model and Issuance Plan

MESA is a native token issued by MetaVisa Protocol.

MESA Function

MetaVisa Protocol charges for the operation and storage of identity, credit systems, and smart contracts, thereby realizing economic incentives for nodes and preventing resource abuse. MetaVisa Protocol also encourages users to contribute personal data and address data so as to complete the MetaVisa Protocol identity system and credit evaluation system.

- Used to pay for the service fee for invoking MetaVisa identity system and credit system services, and the tokens are burned when calling. The specific burning method will be detailed in the whitepaper;
- Used to realize the governance of MetaVisa Protocol. Governance includes voting for node elections, changes to the MetaVisa Protocol identity and credit evaluation model, etc.;
- Used to encourage users to provide personal data and address data actively, and to improve the user's personal identity system in Metaverse;
- Used to upgrade the MID visual presentation by consuming MESA;
- Staking MESA can get rewards, and 30% of the net fees will be distributed as staking rewards;
- 36% of the net fees on MetaVisa Protocol go to a VISA buy/burn.

MESA Issuance Plan

- Total supply: 10,000,000,000

	Distribution	Allocation	Details
Community Rewards	25.00%	2,500,000,000	Emitted across 36 months, with halvenings every 6 months
Team and	20.00%	2,000,000,000	6 months cliff since TGE, 15% will be

Consultants			unlocked in the first 2 years and the remaining will be released quarterly in 2 years
Treasury	12.00%	1,200,000,000	Unlock 20%by the 3rd month from TGE, and the remaining released quarterly in 2 years
Strategic Investor Phase I /Seed	10.00%	1,000,000,000	Unlocked 16% from TGE, and the remaining released linearly in 1 year.
Strategic Investor Phase II /Private	10.00%	1,000,000,000	Unlocked 16% from TGE, and the remaining released linearly in 1 year.
Partnership & Ecosystem Incentive	8.00%	800,000,000	Locked for one year after TGE, and then released linearly in two years.
Marketing	14.00%	1,400,000,000	No lock, but it will not be used in the primary market
IDO	1.00%	100,000,000	100% unlocked.

Table4

Roadmap

Q4 2021

- Initial blockchain integrations: Ethereum
- Data cleaning & algorithm model building
- Data provider integrations
- MetaVisa Protocol V1 testnet

Q1 2022

- MetaVisa Protocol V1 mainnet
- Ethereum Layer 2 blockchain integrations
- Data provider integrations
- DeFi integrations
- NFTs and delivery channels integrations

Q2 2022

- Multi-chain Support: Binance Smart Chain, Huobi Eco Chain, Solana, Polkadot
- Smart Templates: Pre-built Smart Triggers for specific use cases
- MetaVisa Protocol V2 testnet

Q3 2022

- MetaVisa Protocol V2 mainnet
- Identity & Credit Oracle Engine

References

- Durand D. Risk elements in consumer instalment financing[J]. Journal of Marketing,1942, 6(4): 407-408.
- Hand D J ,Henley W E. Statistical classification methods in consumer credit scoring:a review[J]. Journal of the Royal Statistical Society :Series A(Statistics in Society),1997 , 160(3):523-541.
- Hardy W E, Adrian J L. A linear programming alternative to discriminant analysis in credit scoring[J]. Agribusiness , 1985, 1(4): 285-292.
- Rosenberg E,Gleit A. Quantitative methods in credit management :a survey[J]. Operations research ,1994,42(4): 589-613.
- Chen H , Tiňo P , Yao X. Predictive ensemble pruning by expectation propagation[J]. IEEE Transactions on Knowledge and Data Engineering , 2009,21(7): 999-1013.
- Fahey B. A critical review of neoclassical modeling techniques in structured finance[J]. Journal of Post Keynesian Economics , 2013, 35(3): 319-340.
- Martens D,Vanthienen J ,Verbeke W. Performance of classification models from a user perspective[J]. Decision Support Systems 2011 ,51(4):782-793.
- Lee T S,Chiu C C ,Chou Y C. Mining the customer credit using classification and regression tree and multivariat e adaptive regression splines[J]. Computational Statistics & Data Analysis , 2006, 50(4): 1113-1130.
- Orgler Y E. A credit scoring model for commercial loans[J]. Journal of Money , Credit and Banking , 1970,2(4): 435-445.
- Fitzpatrick T , McQuinn K. House prices and mo rtgage credit : Empirical evidence for Ireland[J]. The manchester school ,2007, 75(1): 82-103.
- Orgler Y E. A credit scoring model for commercial loans[J]. Journal of Money , Credit and Banking , 1970,2(4): 435-445.
- Demiroglu C ,James C. The use of bank lines of credit in corporate liquidity management: A review of empirical evidence[J]. Journal of Banking & Finance , 2011, 35 (4):775-782.
- Khashman A. A neural network model for credit risk evaluation[J]. International Journal of Neural Systems ,2009, 19(4): 285-294.

- Hartigan J A, Wong M A. Algorithm AS 136: A k -means clustering algorithm [J]. Journal of the Royal Statistical Society , 1979,28(1):100-108.
- Henley W E ,Hand D J. A k -nearest-neighbour classifier for assessing consumer credit risk[J]. The Statistician , 1996: 77-95.
- Yeh I C , Lien C. The comparisons of data mining techniques for t he predictive accuracy of probability of default of credit card clients[J]. Expert Systems with Applications ,2009, 36(2): 2473-2480.
- Lomax S , Vadera S. A survey of cost -sensitive decision tree induction algorithms[J]. ACM Computing Surveys (CSUR),2013 , 45(2): 16.
- Pradhan B. A comparative study on the predictive ability of the decision tree ,support vector machine and neuro -fuzzy models in landslide susceptibility mapping using GIS[J]. Computers & Geosciences ,2013, 51: 350-365.
- Makowski P. Credit scoring branches out [J]. Credit World ,1985,75(1):30-37.
- Olaru C, Wehenkel L. A complete fuzzy decision tree technique[J]. Fuzzy sets and systems , 2003, 138(2): 221-254.
- Nie G,Rowe W,Zhang L. Credit card churn forecasting by logistic regression and decision tree[J]. Expert Systems with Applications ,2011, 38 (12): 15273-15285.
- Stegaroiu C E,Stegaroiu V. The algorithm for the development of global financial crises[J]. African Journal of Business Management , 2010, 4 (19): 4183-4190.
- Paass G, Kindermann J. Bayesian classification trees with ov erlapping leaves applied to credit -scoring[C]. Pacific -Asia Conference on Knowledge Discovery and Data Mining. Melbourne : Springer Berlin Heidelberg , 1998: 234-245.
- Giudici P. Bayesian data mining , with application to benchmarking and credit scoring[J]. Applie d Stochastic Models in Business and Industry ,2001,17(1): 69-81.
- Weber P , Medina-Oliva G , Simon C. Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas[J]. Engineering Applications of Artificial Intelligence ,2012, 25(4): 671-682.
- Baesens B , Egmont-Petersen M , Castelo R. Learning Bayesian network classifiers for credit scoring using Markov Chain Monte Carlo search[C]. 16th International Conference on Pattern Recognition ,Quebec City:IEEE ,2002, 3: 49-52.
- Migueis V L,Benoit D F,Van den Poel D. Enhanced decision support in credit scoring using Bayesian binary quantile regression[J]. Journal of the Operational Research Society,2013 , 64(9):1374-1383.
- Orlovs A,Braslins G. Cluster impact on company creditworthiness : case of Latvia[J]. Economics and Management , 2013,18(1): 68-76.
- Zhang D Q , Chen S C. Clustering incomplete data using kernel -based fuzzy c-means algorithm[J]. Neural Processing Letters ,2003,18 (3):155-162.
- Soni H N,Joshi M. A fuzzy framework for coordinating pricing and invento ry policies for deteriorating items under retailer partial trade credit financing[J].
- Computers & Industrial Engineering , 2013 ,66(4): 865-878.
- Wu K L, Yang M S. Alternative c -means clustering algorithms[J]. Pattern recognition, 2002,35 (10):2267-2278.

- De Andres J , Lorca P , de Cos Juez F J. Bankruptcy forecasting : A hybrid approach using Fuzzy c -means clustering and Multivariate Adaptive Regression Splines[J]. Expert Systems with Applications , 2011, 38 (3): 1866-1875.
- Vojtek M, Koeenda E. Credit -scoring methods[J]. Journ al of Economics and Finance , 2006, 56 (3-4):152-167.
- Mangasarian O L. Linear and nonlinear separation of patterns by linear programming[J]. Operations research , 1965 , 13(3): 444-452.
- Baesens B, Van Gestel T, Viaene S, et al. Benchmarking state -of-the-art classification algorithms for credit scoring[J]. Journal of the operational research society, 2003,54(6): 627-635.
- Duda R O, Hart P E, Stork D G. Pattern classification[M]. New York ;John Wiley & Sons, 2012 : 215-281.
- Nath R,Jackson W M ,Jones T W. A comparison of the cla ssical and the linear programming approaches to the classification problem in discriminant analysis[J]. Journal of statistical computation and simulation ,1992 ,41(1-2): 73-93.
- Shi Y, Peng Y , Xu W. Data mining via multiple criteria linear programming : applications in credit card portfolio management[J]. International Journal of Information Technology & Decision Making , 2002, 1(1):131-151.
- İç YT. Development of a credit limit allocation model for banks using an integrated Fuzzy TOPSIS and linear programming[J]. Exp ert Systems with Applications , 2012,39(5): 5309-5316.
- Abdou H A, Pointon J. Credit scoring , statistical techniques and evaluation criteria: A review of the literature[J]. Intelligent Systems in Accounting , Finance and Management ,2011, 18(2-3): 59-88.
- Bahrammirzaee A. A comparative survey of artificial intelligence applications in finance: artificial neural networks , expert system and hybrid intelligent systems[J]. Neural Computing and Applications , 2010, 19(8): 1165-1195.
- Khashman A. Credit risk evaluation using neural networks:Emotional versus conventional models[J]. Applied Soft Computing ,2011 ,11(8):5477-5484.
- West D. Neural network credit scoring models[J]. Computers & Operations Research, 2000, 27(11):1131-1152.
- Abdou H,Pointon J ,El-Masry A. Neural nets versus conventi onal techniques in credit scoring in Egyptian banking[J]. Expert Systems with Applications , 2008,35(3): 1275-1292.
- Khashei M, Rezvan M T , Hamadani A Z. A bi - level neural - based fuzzy classification approach for credit scoring problems[J]. Complexity , 2013, 18 (6): 46-57.
- Che L, Hwang C J , Ni C. Research on SVM Distance based Credit Scorin g Model for Network Management[J]. Journal of Convergence Information Technology, 2012 , 7(1):476-482.
- Ong C S,Huang J J ,Tzeng G H. Building credit scoring models using genetic programming[J]. Expert Systems with Applications ,2005,29(1): 41-47.
- Desai V S, Conway D G, Crook J N, et al. Credit -scoring models in the credit-union environment using neural networks and genetic algorit hms[J]. IMA Journal of Management Mathematics , 1997, 8(4):323-346.

- Abdou H A. Genetic programming for credit scoring : The case of Egyptian public sector banks[J]. Expert Systems with Applications , 2009,36 (9): 11402-11417.
- Qing-Shan C , De-Fu Z , Li-Jun W. A modifie d genetic programming for behavior scoring problem[C]. IEEE Symposium on Computational Intelligence and Data Mining. Honolulu : IEEE ,2007: 535-539.
- Bates J M, Granger C W J. The combination of forecasts[J]. Journal of the Operational Research Society,1969 , 20(4):451-468.
- Clemen R T. Combining forecasts : A review and annotated bibliography[J]. International journal of forecasting , 1989, 5 (4): 559-583.
- Kuncheva L I,Whitaker C J ,Shipp C A. Is independence good for combining classifiers[C]. 15th International Conference on Pattern Recognition. Barcelona : IEEE , 2000,2: 168-171.
- Siami M,Gholamian M R ,Basiri J. An application of locally linear model tree algorithm with combination of feature selection in credit scoring[J]. International Journal of Systems Science , 2014,45(10):2213-2222.