Webology, Volume 18, Number 4, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I4/103

Decision Tree Algorithm for Credit Card Fraud Detection

[1]Aditya Joshi, [2]Anuj Singh, [3]Shikha Chauhan, [4]Anupama Sharma

[1]Department of Computer Science and Engineering, Graphic Era Deemed to be University,
Dehradun, Uttarakhand, India,
[2]Department of Computer Science and Engineering, Graphic Era Deemed to be University,
Dehradun, Uttarakhand, India
[3]Department of Computer Science and Engineering, Graphic Era Deemed to be University,
Dehradun, Uttarakhand, India
[4]Research Scholar, School of Management Studies, Graphic Era University, Dehradun

ABSTRACT

In the financial world, new business-making systems have emerged as technology advances. The credit card system is one of them. However, there are many loopholes in this system, which caused a lot of problems with this system as a way of credit card fraud. As a result, both the industry and customers who use credit cards are losing a lot of money. The purpose is to detect fraud in the credit card industry with the use of a machine learning system. In this case, the Decision Tree method is employed for fraud detection. Using some public data as samples, the model's effectiveness may be determined. Following that, we look into a set of real-world credit card data from banking organizations. In addition to this, some clutter is added to the data sample to assist in confirming the robustness of the system. The method's relevance is that it creates a tree for the user's activity and then uses that tree to discover fraud transactions. The findings absolutely show that mainstream selection technologies have achieved considerable accuracy in detecting credit card fraud situations.

Keywords: Business-making systems, Decision tree, Information Gain, Fraud detection, Credit Card

INTRODUCTION

A Fraud means an intentional act that is done for some profit, mainly monetary gain. It is an unethical act whose incidence is increasing on a daily basis [1]. It is defined as a criminal fraud intended to acquire financial or personal benefit. Fraud prevention and detection systems are two of the most important strategies for preventing fraud and losses caused by fraudulent activity [2]. Fraud prevention is a proactive strategy aimed at preventing fraud from occurring. When criminals bypass fraud protection systems and initiate a fraudulent transaction, fraud detection systems come in. Credit card fraud is on the rise as the use of electronic payment systems such as debit and credit cards grows. This card can be used for payment in both online and offline mode. For online payment modes, there are cases where you do not need to physically present your card. In this case, card data is susceptible to attack by cybercriminals and hackers. This kind of scam results in losing millions each and every year. Many algorithms have been developed to overcome this problem. In order to most efficiently solve this problem, various detection methods are being considered. Digital payments are very common nowadays, but they come according to their own array of challenges. There are numerous issues that arise during fraudulent searches. The process of accepting or rejecting a transaction occurs within a very short time ranging from microseconds to milliseconds. Therefore, the process used to detect illegal transactions must be very quick and efficient. Another issue is the large number of concurrent transactions of the same type. This renders it challenging to observe each transaction separately and to identify cheating. As a result, an effective fraud detection system must be implemented in order to distinguish between legitimate and illegal transactions. The paper has its main objective to use Decision Tree model of machine learning to evaluate an imbalanced dataset.

EXISTING SYSTEM

Since, credit card fraud detection (CCFD) systems are an advanced researched area, there are various algorithms and techniques for implementing these systems. One of the earliest systems is the CCFD system using the Markov's model [3]. Various other existing algorithms used in credit cards such as Support vector machine fraud detection systems include cost-sensitive decision trees (CSDT), random forest and more [4][5]. CCFD is also proposed using neural networks [6]. Existing systems using neural network get recognition value according to whale swarm optimization algorithm, it uses the Back propagation network for changing the value where the errors were detected [7]. Studies showed the use of GA Feature Selection on Naïve Bayesian Random Forest and SVM for detecting fraudulent transactions [8]. The research study elaborated on Sequential Behavior Information Processing Using Deep Learning as well as the Markov Transition Field in Online Fraudulent Activities [9]. A method named Attributed Sequence Embedding was displayed, in which various data sets are created using the process [10]. All these techniques have significant drawbacks, such as reduced levels of accuracy, and inefficiency, sometimes categorized as buying regular transactions, and vice versa. The objective of this paper is to find out a new method for detecting fraud, increasing the accuracy and less complexity and time for results. The data set in this paper is based on actual transaction data of European Company, the privacy of which is treated as confidential.

PROPOSED WORK

Decision tree technique is statistical data mining technique in which independent and dependent properties are logically expressed in a structure in the form of a tree illustrated in Fig. 1. The categorization rules derived from the decision tree are if then expressions, and to generate each rule, all tests must pass. Decision trees usually split a complex problem into many simple ones, and use iteration to solve sub- problems. The tree is a predictive decision support tool that creates mappings of possible outcomes from different observations. There are numerous prominent classifiers for generating class models from decision trees. To improve precision and avoid overfitting, During the pruning step, such classifiers create a decision tree and afterwards clean up subtrees from the decision tree. This tree can be created by applying machine learning algorithms to the credit card database, such as ID3, C4.5, and multi-layer pruned classifier (MLPC). The aim of the Decision Tree model is to build a small decision tree with high precision. Based on credit card fraud detection, the decision tree has two stages. The initial step is to build a decision tree using the training data provided, and the later step is to use decision rules to classify incoming transactions. The decision tree's input data is labelled with class labels, such as legitimate or fraudulent. The system monitors each account individually using appropriate descriptors to identify transactions and flags as legitimate or legitimate. In the course of Decision Tree depicted in Fig. 2, all training examples start with one node representing the tree data set at the root node. Each node is split into child nodes in a method-specific binary or multipartition fashion. The decision rules are read one by one from the decision table for each transaction that you classify as Match the transaction fields to each decision rule. It first finds an exact match and indicates the matched rule and transaction class of that class. If no match is found, the highest risk among matching rules is selected and the transaction class is populated with the matched rules of that class. This indicates if a new transaction is a fraud of the same form, The node has been renamed the leaf and is flagged as fraudulent. This model is both quick and adaptable. The MLPC approach is utilized as pre-pruning, which stops the tree's growth at the pruning level specified before construction. It consists of a tree-top-down recursive partitioning and conquest method. Initially all training examples are maintained on the route. The sample is then recursively split based on the chosen attributes. As the entropy metric, choose the split attribute. Repeat the necessary stages until any of the four conditions is met:

1. All samples from a given node pertain to the same class.
2. There are also no other properties for partitioning.
3. There are no remaining sample.
4. Prune level is achieved as set.

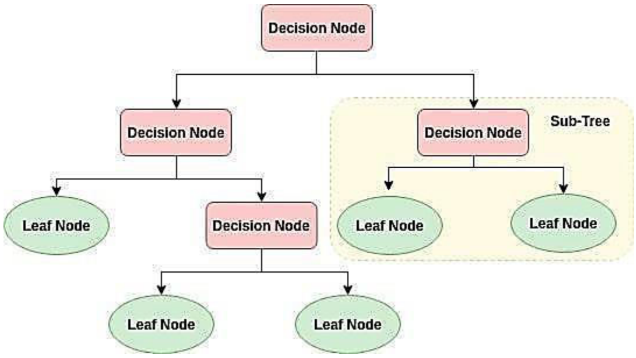

Figure 1. Decision Tree Architecture

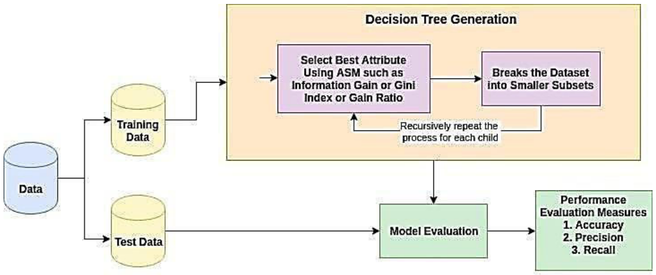
Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Training Data
Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Figure 2. Decision Tree Flow Diagram

RESULT AND ANALYSIS

Measures for selecting attribute

An empirical method for selecting the best division criterion for dividing data is attribute selection measurement (ASM) [11]. Because it facilitates in the determination of breakpoints for a set of nodes, it's also known as a split rule. ASM describe the specified data set and provide a ranking for each feature or attribute. The high score attribute is selected in the split attribute source. For continuous value attribute, you need to define the split point for the branch. The most common selection criteria are gain ratio, Gini's coefficient and information gain.

IG (Information Gain)

Entropy's concept was invented by Shannon [12], it measures the impurities of input set. Entropy is also known as randomness or the imperfections in a system in mathematics and science. It corresponds to the impurities in a collection of various examples in information theory. As information gain takes place, the entropy value will decrease. The difference among entropy prior to actually partitioning and average entropy after the partitioning of a dataset based on a number of attribute values is calculated using information gain. The ID3 decision tree algorithm, which stands for Iterative Dichotomiser Decision Tree Algorithm, uses the IG value, or information gain value.

$$Inf(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad (1)$$

Here, the probability that any tuple in D belongs to class *Ci* is *Pi*.

$$Inf_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Inf(D_j) \qquad (2)$$

$$G(A) = Inf(D) - Inf_A(D) \qquad (3)$$

Where,

Inf(D) is the average amount of information needed to identify a tuple's class label in D..

|Dj| / |D| act as the jth partition weight.

$Inf_A(D)$ is likely to be required to classify tuples from D according to A's partitioning.

Attribute A having the more information gain, G(A), is chosen as partitioning attribute at Nth Node.

Gain Ratio

The benefits of information are they are biased for attributes that use many results. This means that unique values prefer many attributes. For example, consider the case where the information (D) is 0 due to a pure partition on an attribute that has a unique identifier such as a customer ID. This maximizes the information gain and creates unnecessary partition. ID3's improved C4.5 uses an extension of information gain known for his benefit percentage. The gain ratio addresses the issue of bias by using the partitioning information to normalize the information gain value. J48 is the Java implementation of the C4.5 algorithm.

$$SplitInf_A(D) = -\sum_{j=1}^{v} \frac{|D_i|}{|D|} \times \log_2(\frac{|D_i|}{|D|}) \qquad (4)$$

Here,

|Di| / |D| act as the ith partition weight.

v is attribute A’s discrete number value.

The gain ratio is defined as

$$GR(A) = \frac{G(A)}{SplitInf_A(D)} \qquad (5)$$

Where GR = Gain Ratio
G = Gain
A = Attribute
D = Information

The highest gain ratio attribute is chosen as the partitioning attribute or Source.

Information Gain Pseudocode

```
IG(attribute, example, EOS)
GV = EOS
for value in AV(examples, attribute):
SV = subset(value, attribute, examples)
GV= GV- (number in SV)/(number of examples) * entropy(SV)
return GV
```

Where, IG =Info Gain, EOS= Entropy of Set GV=Gain Value AV=Attribute values, SV=Sub Value

Entropy Pseudocode

```
entropy(example)
……
log2(y) = log(y) / log(2)
……
     result = 0
dict= summarizeExamples(targetAttribute, examples) for key in dict:
proportion = dict[key]/number of total examples
result =result - proportion (log2(proportion))
return result
```

Fig. 3 shows the confusion matrix of prediction results obtained by applying a decision tree algorithm of machine learning. In this, you can see that the decision tree algorithm is correctly predicting zeros in the final output 94739.0 times and incorrectly predicting zeros 30.0 times. The decision tree algorithm makes 130 accurate predictions and 37 incorrect predictions.

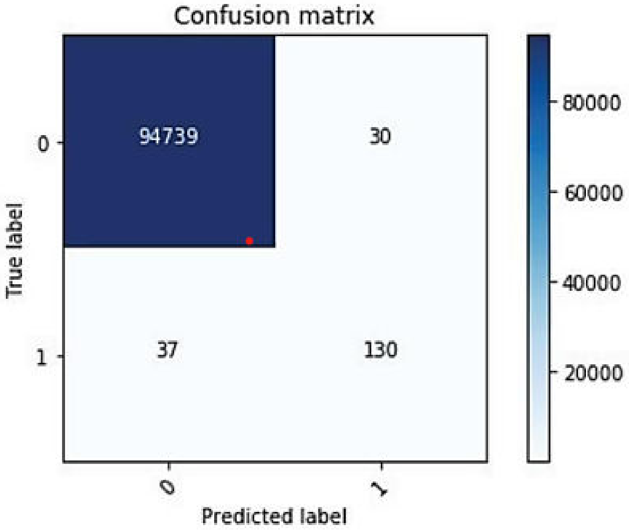

Figure 3. Confusion Matrix of prediction results

CONCLUSION

In this study, we used Kaggle's credit card dataset to validate the efficacy of various supervised machine learning models in predicting the likelihood of fraudulent transactions. To reach a specific outcome, we used accuracy, sensitivity, and time as determinants. Accuracy is not used as an attribute since it is not sensitive to class imbalance and does not provide a clear answer. KNN, Naive Bayes, Decision Tree, K - means, and Random Forest models were investigated. We have predicted that the best-suited model that not only has accuracy but is time sensitive is the Decision tree classification model. Despite the fact that the analysis demonstrates that the Random Forest model has a slightly higher sensitivity than the Decision Tree model, However, we chose the decision tree over the random forest because the random forest takes an extremely long time to test the data. Decision trees are the recommended model for negative detection because predictions need to take the least amount of time.

REFERENCES

1. V. Dheepa, R. Dhanapal, “Analysis of Credit Card Fraud Detection Methods”, International Journal of Recent Trends in Engineering, Vol. 2, no. 3, pp 126 – 128, 2019.
2. S. Dzomira, “Fraud prevention and detection,” Prevention, vol. 6, no. 14, 2015.
3. V. Bhusari and S. Patil, “Study of Hidden Markov Model in Credit Card Fraudulent Detection”. International Journal of Computer Applications, Vol. 20, no. 5, pp 33 – 36, 2011.
4. Y. Sahin, E. Duman, “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines”, Proceeding of International Multi-Conference of Engineering and Computer

Statistics, Vol. 1, 2011.

5. Y. Sahin, S. Bulkan, and E. Duman, “A cost-sensitive decision tree approach for fraud detection,” Expert Systems with Applications, vol. 40, no. 15, pp. 5916–5923, 2013.
6. J. Zhing and A.A. Ghorbani, “Improved competitive learning neural network for network intrusion and fraud detection”, Neurocomputing, vol.75, no. 1, pp. 135-145, 2012.
7. C. Wang, Y. Wang, Z. Ye, L. Yan, W. Cai, and S. Pan, “Credit card fraud detection based on whale algorithm optimized bp neural network,” in 2018 13th international conference on computer science & education (ICCSE). IEEE, 2018, pp. 1–4.
8. Y. K. Saheed, M. A. Hambali, M. O. Arowolo, and Y. A. Olasupo, “Application of ga feature selection on naive bayes, random forest and svm for credit card fraud detection,” in 2020 international conference on decision aid sciences and application (DASA). IEEE, 2020, pp.1091–1097.
9. R. Zhang, F. Zheng, and W. Min, “Sequential behavioral data processing using deep learning and the markov transition field in online fraud detection,” arXiv preprint arXiv:1808.05329, 2018.
10. Z. Zhuang, X. Kong, R. Elke, J. Zouaoui, and A. Arora, “Attributed sequence embedding,” in 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019, pp. 1723–1728.
11. L. A. Badulescu, “The choice of the attribute selection measure in decision tree induction,” Annals of the University of Craiova-Mathematics and Computer Science Series, vol. 34, pp. 88–93, 2007.
12. B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, “A comparative study of decision tree id3 and c4. 5,” International Journal of Advanced Computer Science and Applications, vol. 4, no. 2, pp. 13–19, 2014.