CREDIT CARD FORGERY ANALYSIS

[1]Dr.G.Jayamurugan MCA., M.E., Ph.D , [2]Gausick.G, [3]Subash.P,

[1]Associate Professor, Department of Computer Engineering,

Sengunthar Engineering College (AUTONOMOUS).

Department of Computer Engineering,

Sengunthar Engineering College (AUTONOMOUS).

Tiruchengode – 637 205

1NTRODUCTION

Now a day the usage of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising.Online Shopping – one of the largest and fast going trendMode of payment – credit card, debit card, Net Banking Online payment does not require physical card Major Risk – credit card detail is known to other

2 RELATED WORK

2.1 A Cost-Sensitive Decision Tree Approach For Fraud Detection

Author :Yusuf Sahin a,, Serol Bulkan b, Ekrem Duman c

With the developments in the information technology, fraud is spreading all over the world, resulting in huge financial losses. Though fraud prevention mechanisms such as CHIP&PIN are developed for credit card systems, these mechanisms do not prevent the most common fraud types such as fraudulent credit card usages over virtual POS (Point Of Sale) terminals or mail orders so called online credit card fraud. As a result, fraud detection becomes the essential tool and probably the best way to stop such fraud types. In this study, a new cost-sensitive decision tree approach which minimizes the sum of misclassification costs while selecting the splitting attribute at each non-terminal node is developed and the performance of this approach is compared with the well-known traditional classification models on a real world credit card data set. In this approach, misclassification costs are taken as varying. The results show that this cost-sensitive decision tree algorithm outperforms the existing well-known methods on the given problem set with respect to the well-known performance metrics such as accuracy and true positive rate, but also a newly defined cost-sensitive metric specific to credit card fraud detection domain. Accordingly, financial losses due to fraudulent transactions can be decreased more by the implementation of this approach in fraud detection systems.

3 SYSTEM ANALYSIS

3.1EXISTING SYSTEM

Three methods to detect fraud are presented. Firstly, clustering model is used to classify the legal and fraudulent transaction using data clusterization of regions of parameter value. Secondly, Gaussian mixture model is used to model the probability density of credit card user's past behavior so that the probability of current behavior can be calculated to detect any abnormalities from the past behavior. Lastly, Bayesian networks are used to describe the statistics of a particular user and the statistics of different fraud scenarios. The main task is to explore different views of the same problem and

Copyright © Author

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

see what can be learned from the application of each different technique.

DISADVANTAGES

- The high amount of losses due to fraud and the awareness of the relation between loss and the available limit has to be reduced.
- Testing credit card FDSs using real data set is a difficult task.
- The fraud has to be deducted in real time and the number of false alert

3.2PROPOSED SYSTEM

Total of twelve machine learning algorithms are used for detecting credit card fraud. The algorithms range from standard neural networks to deep learning models. They are evaluated using both benchmark and real world credit card data sets. In addition, the AdaBoost and majority voting methods are applied for forming hybrid models. To further evaluate the robustness and reliability of the models, noise is added to the real-world data set. The key contribution of this paper is the evaluation of a variety of machine learning models with a real-world credit card data set for fraud detection.

ADVANTAGES

- The system is very fast due to AdaBoost Technique.
- Effective Majority Voting techniques.
- Easily Detect credit card fraud detection.

4 SYSTM TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES *OF* TESTS

Unittesting

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by

Copyright © Author

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SystemTest

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WhiteBoxTesting

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BlackBoxTesting

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design of the project is turned into a working system. It is a stage where the operation of the system is monitored to ensure that it continues to work effectively. Education and training of the users are also essential to ensure smooth functioning of the system.

The major tasks involved in the implementation are

- Computer based/system testing.
- Training the user personnel
- Full system testing and making the necessary changes as desired by the user.
- Change over.
- Maintenance.

The implementation strategy used is the parallel changeover. The automated system has been put to use gradually so that its usage can prove better for the concern. After the system has been tested, the implementation type or the change over technique

Copyright © Author

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

from the existing system to the new system is a step-by-step process. In the system, at first only a module of the system is implemented and checked for suitability and efficiency. When the end-user related to the particular module is satisfied with the performance, the next step of implementation is preceded.

Implementation to some extent is also parallel. For instance, modules, which are not linked, with other modules are implemented parallel and the remaining is the step-by-step process. Backups are necessary since any time unexpected events may happen. And so during the program execution, the records are stored in the workspace. This helps to recover the original status of the records from any accidental updating or intentional deletion of records.

IMPLEMENTATION PROCEDURES

Implementation means converting older system to a new design in operation. This involves creating computer capable files and basic software needed to run this system. The basic concept for implementation needed is software installation and system requirements. So in order to implement them, suitable hardware and software must be available.

5 CONCLUSION AND FUTURE ENHANCEMENTS

5.1 CONCLUSION

In this paper, we have performed several machine and deep learning models to detect whether an online transaction is legitimate or fraud on the IEEE-CIS Fraud Detection dataset as well built our model which is BiLSTM-MaxPooling-BiGRUFigure MaxPooling that based on bidirectional LSTM and GRU. We also tested several methods to deal with highly imbalanced datasets including undersampling, oversampling and SMOTE. Set of evaluation metrics used to evaluate the performance of the models. The results from machine learning classifiers show that the best AUC was 80% and 81% that achieved by hard voting with undersampling and oversampling technique. However, the results from machine learning classifiers were not promising compared with our model that achieved 91.37% AUC.

5.2 FUTURE WORK

For future work, the methods studied in this paper will be extended to online learning models. In addition, other online learning models will be investigated. The use of online learning will enable rapid detection of fraud cases, potentially in real-time. This in turn will help detect and prevent fraudulent transactions before they take place, which will reduce the number of losses incurred every day in the financial sector.

REFERENCES

[1] S. Gaur, “Bringing context awareness to iot-based wireless sensor networks,” in PerCom’15. IEEE, 2015.

[2] A. Solanas, C. Patsakis, M. Conti, I. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. Perez-martinez, R. Di Pietro, D. Perrea, and A. Martnez-Balleste, “Smart health: a context-aware health paradigm within smart cities,” IEEE Communications Magazine, vol. 52, no. 8, pp. 74–81, 2014.

[3] Y. Wang, G. Attebury, and B. Ramamurthy, “A survey of security issues in wireless sensor networks,” IEEE Communications Surveys & Tutorials, vol. 8, no. 2, pp. 2–23.

[4] O. Garcia-Morchon, S. Keoh, S. Kumar, R. Hummen, and R. Struik, “Security considerations in the ip-based internet of things,” 2012. [Online]. Available: https://tools.ietf.org/html/ draft-garcia-core-security-04

Copyright © Author

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

[5] M. Conti, R. Di Pietro, and A. Spognardi, “Clone wars: Distributed detection of clone attacks in mobile wsns,” Journal of Computer and System Sciences, vol. 80, no. 3, pp. 654–669, 2014.

[6] M. Conti, “Clone detection,” in Secure Wireless Sensor Networks. Springer, 2016, pp. 75–100.

[7] A. K. Mishra and A. K. Turuk, “A comparative analysis of node replica detection schemes in . 1022–1034, 2012.

wireless sensor networks,” Journal of Network and Computer Applications, vol. 61, pp. 21–32, 2016.

[8] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, “Detecting node replication attacks in wireless sensor networks: a survey,” Journal of Network and Computer Applications, vol. 35, no. 3, pp

Copyright © Author

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731