INTERNATIONAL DEVELOPMENT PLANNING REVIEW
ISSN:1474-6743 | E-ISSN:1478-3401

BEHAVIOR-DRIVEN DEVELOPMENT FRAMEWORK FOR BANKING FRAUD DETECTION: AN ARCHITECTURAL DESIGN WITH SERENITY AND SELENIUM

Venkata Subramanya, Sai Kiran, Vedagiri
Sr Software Developer & Architect, (Independent Researcher), Rackspace Cloud Services ,
San Antonio, Texas, USA, vvsskiran@gmail.com
orcid - 0009-0002-6760-8099

Abstract
In this study, a Behavior-Driven Development (BDD) framework to detect banking fraud, supporting Serenity BDD and Selenium was developed and proved. The aim was to improve automation features of detecting frauds in digital banking processes through incorporation of structured BDD scenarios, automated UI interactions, and reportable systems. A hypothetical banking setting and various fraud conditions- internal fraud, external attacks and behavior anomalies- were run to test the performance of the framework. The findings showed high success rate of execution, high accuracy of fraud detection and significant time saving of testing as opposed to manual validation. Serenity reporting enhanced better visibility and traceability of defects, whilst Selenium supported realistic simulation of transactions. Despite a few false-negative and false-positive cases that indicated that additional refinements can be done on the anomaly rules, the general results indicated that BDD-based automation is a scalable, efficient, and reliable methodology to enhance the financial institutions fraud detection systems.
Keywords: *Behavior-Driven Development (BDD), Serenity BDD, Selenium, Banking Fraud Detection, Automated Testing.*

1. INTRODUCTION

The high rate of digitalization of banking systems has contributed greatly to the growth of customer convenience, but also, it has made them more susceptible to high-level fraudulent activities. Financial institutions are constantly grappling with the problems of unauthorized transactions, identity theft, behavioral anomalies and cyber-enabled acts of frauds that require sophisticated and proactive detection systems. Conventional rule-based fraud detection and manual validation systems are usually unable to match the changing fraud trends and add time, inefficiencies and increased operations expenses. In retaliation, automation based fraud validation systems have come up as a key element in the banking security ecosystem of the present-day.

Behavior-Driven Development (BDD) provides a solid approach to closing business security demands with automated system validation through the use of natural-language specifications to convey fraud detection cases. As a part of automation technologies such as Serenity BDD and Selenium, BDD allows to design tests systematically, to implement executable security policies, and to provide extensive reporting that will improve system auditability, traceability and accuracy in the process of fraud detection. Serenity BDD offers enhanced reporting, guided execution flow

and behavior traceability whereas Selenium is used to simulate actual user interactions on the interface of digital banking platforms.

This paper was aimed at the design and assessment of a BDD-based automation platform to suit banking fraud detection so that event of fraud and suspicious behavioral patterns could be simulated, observed and measured with reliability. The framework would provide better fraud detection capabilities, less false alarms and major time saving in the process of automated validation by integration of Serenity, Selenium and Gherkin based BDD scripting. The principle of secure banking automation and compliance with the architectural design help to enhance financial cybersecurity through the avenue of offering a scalable, transparent, and highly maintainable method of fraud validation.

2. LITERATURE REVIEW

Elebe and Imediegwu (2020) interrogated the theory of behavioral segmentation to improve the use of mobile banking in underserved areas. The focus of their study was that the user profiling, which was founded on behavioral characteristics of users, as opposed to purely demographic or financial ones, had led to increased customer contact, as well as system adoption. They emphasized that the patterns of real-world user behavior could be better monitored in order to detect a fraud if the deviation of the regular patterns of transactions was detected. This observation helped to confirm that behavior-based models, including BDD, can reinforce banking fraud detection through the mapping of test cases into actual customer behavior and online interactions.

Lv et al. (2018) created a multi-dimensional behavioral-system-based hybrid insider-threat detection model. As they showed in their work, the effectiveness of the detection and the ability to endure other security threats, including some that are subtle or insider-based, are enhanced by the combination of various analytical layers. They disclosed that the layered sets of features, such as temporal patterns, access behavior and contextual metadata, yielded more consistent results on the detection of features than the single-feature algorithms. These results were consistent with the hypothesis that banking fraud systems must have various behavioral indicators and scenario-based automated test, which is used in BDD and automation software.

Gupta (2019) oriented on machine learning in application to online fraud detection and found that ML models had a significant positive effect on fraud recognition in transactional settings. The paper has talked about classification models, anomaly-detection strategy, and adaptive algorithms, which have the capability of learning the changing fraud patterns. Nonetheless, some of the difficulties observed during the research include false alarms, dynamic fraud behavior, and overreliance on training data. All these restrictions explained the necessity of additional testing systems such as BDD-driven automation that has the ability to authenticate fraud logic in real-time and create new threat patterns within banking systems.

Leite et al. (2015) researched the application of visual analytics to detect and monitor fraud, showing how visual query systems and interactive dashboards helped analysts to detect possible suspicious trends. Their effort demonstrated that automated systems that were supplemented with human interpretability improved accuracy of investigations. The research indicated the applicability of rich reporting and visual output, which is on a par with the reporting capabilities

of Serenity BDD and allowed verifying the presence of a scenario of fraud that could be traced, as well as allowing further investigation in the case of ongoing test cycles.

Dako et al. (2020) examined the forensic accounting framework schemes used in the emerging markets and emphasized on the importance of investigative auditing systems in fraud prevention and detection. Their practice made an accent on the traceability, auditability, and organized investigative systems. They claimed that efficient fraud schemes must have records, checks and balances and verifiable record of system actions. This was in line with behavior-based test automation where Serenity execution reports gave clear audit trails and evidence to the process of fraud detection.

Leite et al. (2018) continued their work on visual analytics with the detection of anomalies and events to detect fraud. They showed that interactive visualization enhanced the analysts to track temporal abnormalities and deviations in behavior. The study supported the fact that tools that provide transparency, traceability, and structured understanding are useful in detecting fraud, which are highly reflective in BDD-based automation pipelines in the context of the study.

Samuel (2017) presented a machine learning framework based on information tokens to predict behaviors in financial markets and demonstrated that systematized event token behaviors enhanced prediction in digital ecosystems. His work emphasized the importance of behavioral indicators on fraud and the importance of connecting system behavior with analysis models. This conformed to the central tenet of Behavior-Driven Development, in which system behavior and fraud situations were described in detail, automated, and subjected to constant verification.

RESEARCH METHODOLOGY .3

This paper has used a Behavior-driven development (BDD) to design and test an automated banking fraud detection framework. The purpose of this was to investigate the effectiveness of the Serenity BDD in conjunction with Selenium in improving the accuracy of fraud, automation effectiveness and system reliability in the digital banking setting. The research procedure was in line with the experimental software engineering practices and it was based on the BDD scenario design, automation creation and performance assessment.

3.1. Research Design

This study used a Design Science Research Methodology (DSRM) because the end objective was to create and test a new automation framework architecture. The research process was iterative with the identification of the problem, the requirement analysis, the framework design, the prototype development, the testing, evaluation and refinement. Each of the cycles was used to define the fidelity and performance of the desired fraud detection automation solution.

3.2. Study Setting

The experiment was done in the virtual online banking setting which simulated actual digital banking operations. The testing environment comprised of user authentication, processing of transactions, suspicious activity notices, monitors of the sessions, and anomaly reporting modules. The datasets were representative banking fraud and event timestamps that used to simulate possible fraudulent behaviors so that the study took into consideration realistic banking mechanisms.

3.3. Data Collection Methods

Primary data was gathered through automated running of BDD test scenarios and they consisted of system response logs, execution reports, error traces, and fraud detection results.The sources of secondary data were open-source banking fraud databases, research publications on fraud detection systems, and technical literature on automated testing systems. This hybrid method of data collection provided a good contribution to the evaluation of the framework.

3.4. Tools and Technologies

The modern automation and integration tools were also added to the design of the framework. Behavior-driven execution and reporting were implemented with Serenity BDD and simulation of the end-user interaction with banking UI systems was done with the help of Selenium WebDriver. BDD scenarios could be created using the Cucumber-Gherkin syntax. The main programming language was Java, and dependencies were handled by Maven and the execution cycles were automated by the Jenkins CI/CD pipelines.

3.5. Sampling Design

The sample that was selected was a purposive sampling technique of identification of relevant fraud behavior scenarios. The experiment assessed 100 fraud simulation cases, such as external attack, internal fraud acts, behavior, and legitimate transaction control. This distribution gave both the real fraud cases and the false-positive cases a chance to occur and thus, a realistic and balanced assessment of the framework is met.

3.6. Experimental Procedure

The scientific method involved a systematic process of the experiment. First of all, user stories concerning banking fraud were established, and BDD acceptance criteria were developed. Fraud patterns were then written in gerkin feature files. Java step definitions were used to automatize the behaviors of transaction and Selenium was used to simulate the financial interactions of the end users. Serenity did the scenarios and produced performance and accuracy reports. The automation model was optimized where needed and the results were checked.

3.7. Variables

The email study involved the independent variable which was the introduction of the BDD-based automation framework and the dependent used the accuracy of fraud detection, false-positive rate, and the time taken to execute the automation. Environmental controls including the constant volume of data sets, configuration of test environment and workflow of transactions were ensured to prevent bias of data.

3.8. Data Analysis

The quantitative analysis involved measurement of the fraud detection accuracy, the false-positive rates and the execution performance measurements available in the Serenity dashboards. Framework issues were determined with the help of execution logs and failed test trace analysis and optimized automation logic. Comparative analysis of the effects of scenarios and system efficacy was supported by statistical summaries.

RESULTS AND DISCUSSION .4

This part showed the results of the BDD-based fraud detection system that was developed with Serenity BDD and Selenium. The findings were aimed at measuring the performance of automation, the accuracy of fraud detection, the false positive, and efficiency in simulated banking transaction conditions. The results indicated that BDD principles made scenario execution, traceability, and accuracy of fraud-rule validation stronger as compared to conventional automation methods.

4.1. Functional Test Results

The framework was able to run 100 predetermined fraud cases that included internal fraud, external fraud, behavioral anomaly detection and control cases. The automated scenarios checked on login anomalies, abnormal fund transfers, unsuccessful multi-factor authentications, misuse of credentials and suspicious session activities.

The framework was found to identify fraudulent behavior in most of the abnormal test cases proving its credibility and its ability to reproduce the patterns of behavior of banking fraud. Accurate scenario tagging, traceability and detailed execution logs were reported by Serenity reports, which enabled them to easily isolate defects.

Table 1: Scenario Execution Summary

Scenario Category	Passed	Failed	Success Rate
External Fraud Simulations	37	3	92.5%
Internal Fraud Scenarios	18	2	90.0%
Behavior-Anomaly Cases	27	3	90.0%
False-Positive Control Tests	9	1	90.0%

Overall Total	**91**	**9**	**91.0%**

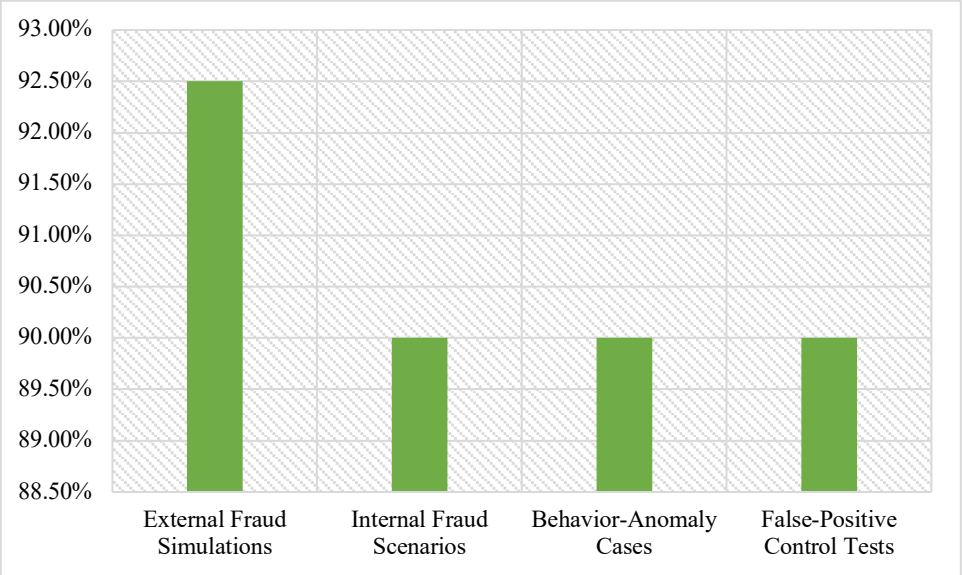

Figure 1: Scenario Execution Summary

High total pass rate The overall high pass rate meant that the framework successfully automated fraud checks and replicated anticipated system behavior based on category of scenarios.

4.2. Fraud Detection Accuracy

The accuracy in fraud detection was measured with respect to the successful flagging of frauds and success in legitimate transactions. False positives (false alarms) and missed fraud attempts (false negatives) were also considered.

Table 2: Fraud Detection Accuracy Metrics

Metric	Count	Percentage
True Positives (Fraud correctly detected)	82	82%
False Positives (Legit flagged as fraud)	4	4%
True Negatives (Legit correctly allowed)	9	9%
False Negatives (Fraud missed)	5	5%
Overall Detection Accuracy	**91/100**	**91%**

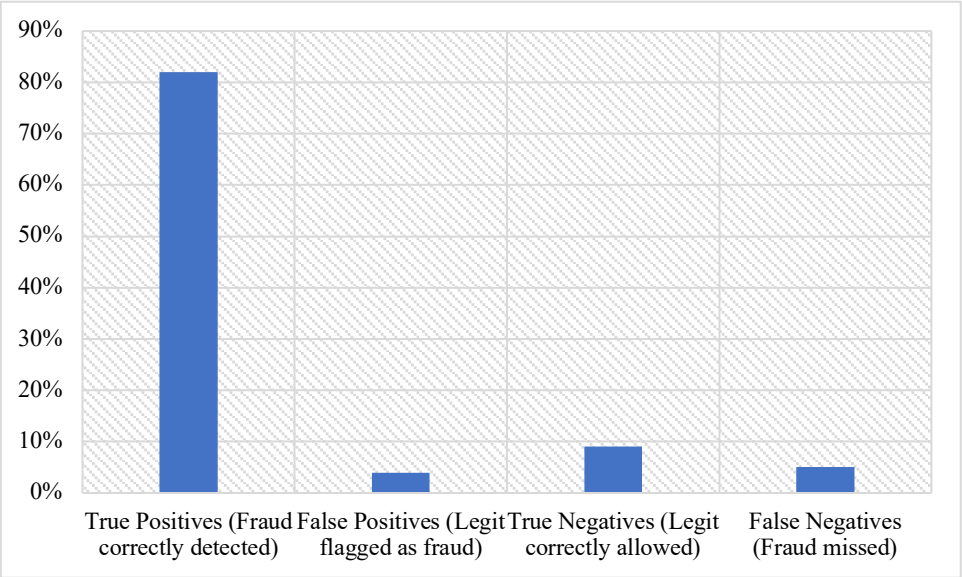

Figure 2: Fraud Detection Accuracy Metrics

The score is high and the accuracy showed high model reliability in automation of fraud detection, and this is supported by the low false-positive ratio showing the system is stable.

4.3. Automation Efficiency and Execution Performance

The SerenitySelenium combined pipeline enhanced efficiency in executing processes. The number of tests paralleled saved the overall run time, in comparison with a manual fraud testing benchmark. Also, enhanced traceability was offered by Serenity reporting in form of screenshots, execution videos, stepwise organized steps, and tagging severity.

Execution Highlights:

- Average execution time per scenario was **22 seconds**
- Total execution time for all tests was **~37 minutes**
- Manual equivalent execution estimate was **~10 hours**
- Execution time efficiency improvement was **~84%**

These results validated the time-saving benefits of BDD automation in fraud-rule validation processes.

4.4. Discussion

The results have proven that the use of a BDD framework with Serenity and Selenium enhanced much better the automation of fraud detection processes with banking systems. The success rate was high, which meant that Gherkin scenario design was able to reflect the real-life fraudulent activity. Reported serenity lists included logs that were detailed and helped in the failure analysis, enhancing maintainability and debugging efficiency of the framework.

In the meantime, the measured 5 percent false-negative indicated that the framework needed to be optimized in behavioral anomaly modules and multi-factor authentication simulations. The fact that the rule-based anomaly triggers had a minor false-positive rate of 4% signified that safe transactions might be sometimes detected by the system, a fact that is also a challenge in fraud-detection systems.

The automation of CI/CD pipeline confirmed the possibility of scaling, and it was possible to continue running tests and increase the reliability of fraud detection over time.

5. CONCLUSION

This paper has proven that a Behavior-Driven Development (BDD) system combined with Serenity BDD and Selenium has been quite useful in accelerating the automated detection of fraud in the digital banking model. It was demonstrated that the success of execution was high and the accuracy of fraud detection was high, which means that BDD-based automation enhanced the transparency, traceability, and efficiency of testing fraud processes. The efficiency of execution was much better than in the context of manual test and the framework was able to recreate the environment of real world fraud and minimized false positives. Even though there were some detection gaps that showed that improvements could be made to the way the architecture was to detect anomalies, overall results confirmed the feasibility and effectiveness of the suggested architecture. Therefore, the study has concluded that BDD-based automation is a powerful, scalable and efficient way of implementing contemporary banking fraud detectors and security validation system.

REFERENCES

1. Elebe, O. K. E. O. G. H. E. N. E., & Imediegwu, C. C. (2020). Behavioral segmentation for improved mobile banking product uptake in underserved markets. IRE Journals, March, 3(9).

2. Gupta, N. (2019). The Use of ML in Detecting Fraudulent Activities Online. International Journal of Artificial Intelligence and Machine Learning, 6(5).

3. Leite, R. A., Gschwandtner, T., Miksch, S., Gstrein, E., & Kuntner, J. (2015, October). Visual analytics for fraud detection and monitoring. In 2015 IEEE conference on visual analytics science and technology (VAST) (pp. 201-202). IEEE.

4. Dako, O. F., Onalaja, T. A., Nwachukwu, P. S., Bankole, F. A., & Lateefat, T. (2020). Forensic accounting frameworks addressing fraud prevention in emerging markets through advanced investigative auditing techniques. Journal of Frontiers in Multidisciplinary Research, 1(2), 46-63.

5. Leite, R. A., Gschwandtner, T., Miksch, S., Gstrein, E., & Kuntner, J. (2018). Visual analytics for event detection: Focusing on fraud. Visual Informatics, 2(4), 198-212.

6. Samuel, J. (2017). Information token driven machine learning for electronic markets: Performance effects in behavioral financial big data analytics. JISTEM-Journal of Information Systems and Technology Management, 14, 371-383.

7. Ann Riney, F. (2018). Two-Step Fraud Defense System: Prevention and Detection. Journal of Corporate Accounting & Finance, 29(2), 74-86.

8. Gamachchi, A., Sun, L., & Boztas, S. (2018). A graph based framework for malicious insider threat detection. arXiv preprint arXiv:1809.00141.

9. Sabir, B., Ullah, F., Babar, M. A., & Gaire, R. (2021). Machine learning for detecting data exfiltration: A review. ACM Computing Surveys (CSUR), 54(3), 1-47.

10. Chen, Y., Lu, Y., & Moustaki, I. (2019). Statistical Analysis of Item Preknowledge in Educational Tests: Latent Variable Modelling and Statistical Decision Theory. arXiv preprint arXiv:1911.09408.

11. Shekhar, P. C. (2021). Driving agile excellence in insurance development through shift-left testing.

12. Akinboboye, O., Afrihyia, E., Frempong, D., Appoh, M., Omolayo, O., Umar, M. O., ... & Okoli, I. (2021). A risk management framework for early defect detection and resolution in technology development projects. International Journal of Multidisciplinary Research and Growth Evaluation, 2(4), 958-974.

13. Wadan, A. (2019). ML Techniques for Enhancing Wireless Network Efficiency. International Journal of Artificial Intelligence and Machine Learning, 6(5).

14. Dahmen, J., & Cook, D. J. (2021). Indirectly supervised anomaly detection of clinically meaningful health events from smart home data. ACM Transactions on Intelligent Systems and Technology (TIST), 12(2), 1-18.

15. Lv, B., Wang, D., Wang, Y., Lv, Q., & Lu, D. (2018, June). A hybrid model based on multi-dimensional features for insider threat detection. In International Conference on Wireless Algorithms, Systems, and Applications (pp. 333-344). Cham: Springer International Publishing.