Design and Development of a Prototype System for Continuous Auditing of Financial Transactions using Machine Learning

Sternford Mavuchi

Information Technology,

Harare Institute Of technology

Harare, Zimbabwe

[sternfordmavuchi@gmail.com](mailto:sternfordmavuchi@gmail.com)

Budwell T Masaiti 2

Information Technology,

Harare institute of technology

Harare, Zimbabwe

[bmasaiti@gmail.com](mailto:bmasaiti@gmail.com)

**Abstract - huge amounts of transactions data are being processed due to ERP systems, which implies that mechanisms should be put in place to monitor and validate transactions continuously without affecting the day to day operations. Continuous Audit system is NRT application that provides the capability of financial data to be checked continuously. This will provide frequent surveillance on the transactions. In any case of an unexpected transaction, it notify the anomaly. CA system can be designed and developed on any platform but for its better functionality we are using machine learning techniques. Machine learning algorithms provide a process of detecting anomalies and generating rules for its detection and prevention. In this paper we are using SVM machine learning algorithm to detect anomalies on real time financial transactions. It picks up anomalies such as overcharges and undercharges of duty on certain goods, redirection of funds, unusual amounts for certain already known transactions. SVM is categorised under supervised algorithms. The study used a sample dataset of transactions from the vehicles importation duty calculations from Zimbabwe Revenue Authority for training and testing the SVM model thereby providing data engineering processes using frameworks such as Apache Kafka to simulate streams of data from the dataset and Apache Spark providing the engine for stream processing of multiple transactions within a short space of time. Apache Spark platform further provide visual distributions for the processed or processed jobs. The prototype showed that SVM algorithm is capable of detecting anomalies with test accuracy of 86% which seem to increase if vast amounts of training data is supplied for data with a lot of features.**

**Keywords: Continuous Audit, NRT (near real time), SVM (Support Vector Machine), supervised, unsupervised, hybrid, Apache Spark, Kafka.**

1. INTRODUCTION

Organisations need to find ways which allow them to inspect their transactions continuously in a way that does not hinder with the ongoing business operations. In this study, emphasis is on building a system that continuously monitor financial transactions in real or near real time basis.

**Continuous Auditing**

It is an electronic auditing system which intend to monitor financial transactions of a firm on a continuous basis. It is a near real time system that analyse and detect anomalies associated with the financial transactions using SVM algorithm. However an anomaly detection system.

Continuous auditing systems are driven by technology to allow automation of error checking and data verification in near real- time. Near Real Time (NRT) system collects transactional data from multiple data sources such as information systems, POS  
machines, Servers and other applications handling financial transactions data. The goal is to monitor and analyse streams of such data using data engineering tools to detect anomalies, possible fraud and corruption, risks of errors and omissions. The continuous data to be analysed is generated from a dataset by means of creating data streams through Apache Kafka, a distribution messaging system that create data streams ready to processed as multiple Kafka topics. During the data analysis and anomaly detection phase, Apache Spark with the help of SVM machine learning algorithm will perform data engineering processes to search and detect anomalies such as risks, errors, omissions, mistakes and fraud associated with the financial data. Therefore present the result through a dashboard and alert responsible personnel in case of non-conformities

1. LITERATURE REVIEW
2. METHODOLOGY

The following diagram shows the architectural design of the proposed prototype of the Continuous Audit. It illustrates the breakdown of the system data pipeline showing clearly the events to be followed from the point of data entry up until data visualisation. The figure also shows the platforms and technologies involved throughout the system.

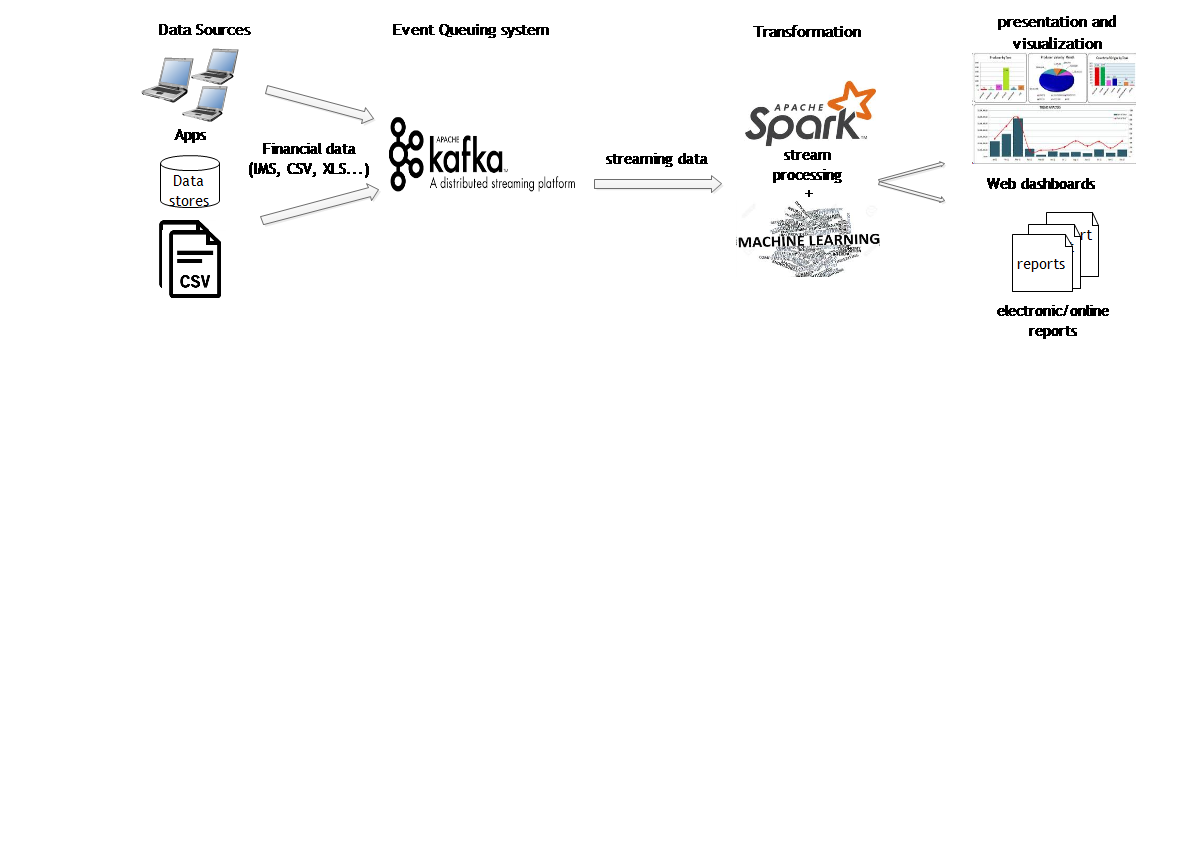


Figure 1: Solution Architecture

Theoretical steps for the proposed methodology:

* Gather training and testing data (vehicles dataset)
* Pre-process both training and testing data removing duplicates, outliers
* Extract features and calculate weight of attributes
* Input for the SVM algorithm is ready.
* Data visualisation

1. DESIGN

**Training and Testing Dataset**

In this research we have created transactional data of sample calculations of duty on imported vehicles at the borders. It is very important to train the Continuous Auditing system with huge amounts of anomaly data.

The vehicles dataset is a huge set of transaction in .csv format.

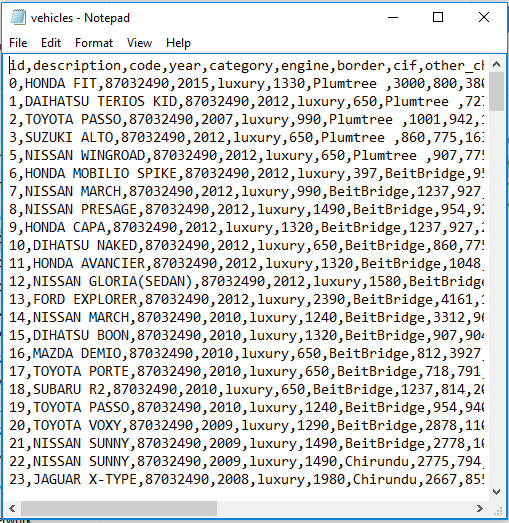


Figure 2: snapshot of vehicles transaction data

**Classification of data**

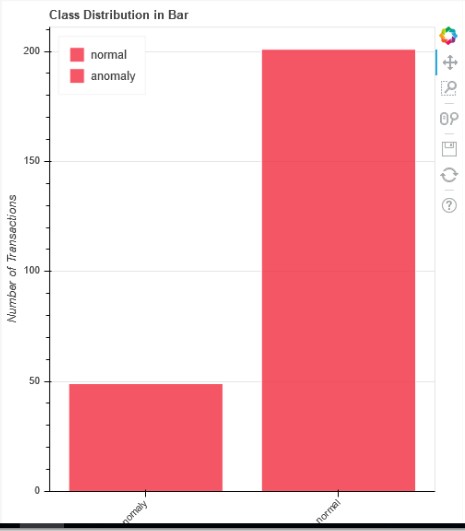
In this study we are using Support vector machine which is a classifying algorithm. Basically divides each transaction into clusters so that they can be easily distinguished according to the particular features representation. The SVM model is build and sits on top of Apache Spark engine and was trained using data from a dataset.



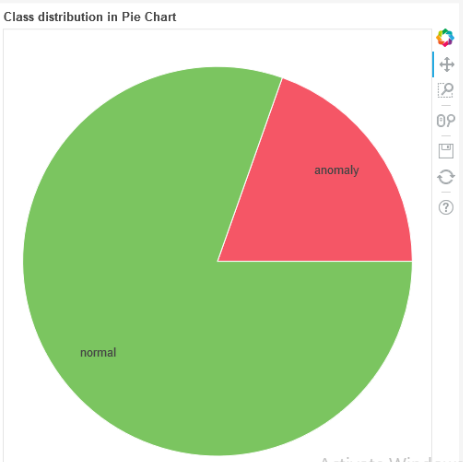
*Figure 3: classification*

**Visualisation of results**

The results of classifications are presented to the user through graphs. Graphs show the distributions of the classes (anomaly and normal transactions)



*Figure 4: Class distribution in Bar Chart*



*Figure 5: Class distribution in Pie Chart*

1. IMPLEMENTATION

**Platforms**

Apache Kafka and spark

Create multiple data streams from the sources of data, ready to be processed as Kafka topics. The Kafka topics are consumed by  
Apache Spark for data analysis and anomaly detection using SVM algorithm. it picks up anomalies such as overcharges and  
undercharges of say duty on certain goods, redirection of funds, unusual amounts for certain already known transactions

Python programing Language

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. This is the language used in the development of the proposed system.

**Frameworks**

Anaconda distribution

Anaconda distribution is an open source data science platform powered by python. It provide over 720 python packages that can be used in the development of any python program. For instance, sci-kit learn, a machine learning package that contains algorithms such as SVM, decision tree, etc.

Django

Django is a high-level python web framework that encourages rapid development and clean, pragmatic design. It follows the model-view-template architectural pattern. In as far as CA is concerned, Django web framework is used as backend of a web dashboard for the continuous monitoring of processing and processed jobs as well as real time visualization

**IDEs**

JetBrains PyCharm community edition

Intelligent Python IDE (instructions developing Editor) with refactoring, debugger, code completion, on-the-fly code analysis and coding productivity orientation. PyCharm integrates well with the above mentioned platforms as well as frameworks that are used to develop the proposed system.

1. DISCUSSION AND RESULTS

The main aim of the study was to detect anomalies in the transaction dataset using algorithms. Literature review showed that algorithms suit for this has applied, but combinations of algorithm are still under study. SVM classification algorithm prove to be the efficient with greater accuracy than most. The system reached a test accuracy of 86%. It shows that the dataset have many features to work with towards the ultimate grouping of transactions which implies that, it require huge amounts of training data in order to raise the accuracy score.

1. FUTURE WORK

The system can be extended to include the ability to continuously learn on its own in case of arrival of new transactional data that haven’t been in the training set.

Furthermore, the proposed prototype, have a 14% chance of misclassifications. However, in case of misclassifications, the classification model can use a predefined set of rules to verify and back up its outcome from prediction.

1. ACKNOWLEDGEMENTS

I wish to acknowledge the assistance received from the following people who made it possible for this document to be put together.

My supervisors, Mr B Masaiti who tirelessly guided me in conducting and compiling the entire project, and my fellow classmates for providing assistance in time of need

1. AUTHOR PROFILES



**Student Author:** Sternford Mavuchi is a fourth-year student at Harare Institute of Technology currently pursuing a BTech Hons degree in Information Technology