Medicare Fraud Detection Using Big Data Analytics

Waasi A Jagirdar, Gautam Vanishree Raghu , Rutvija Deshpande
EECS 6893 , Big Data Analytics
Electrical Engineering, Columbia University

Abstract

Fraud within the Medicare system poses a significant financial burden, costing billions of dollars annually. This project presents a scalable and efficient framework for detecting anomalies and fraudulent claims in Medicare data, leveraging a robust big-data technology stack. Our methodology integrates advanced statistical techniques, such as Z-Score and Interquartile Range (IQR), with machine learning models, including Random Forest and Logistic Regression, achieving up to 99% accuracy in identifying fraudulent claims. The project processes over 9 million rows of Medicare claim data using Apache PySpark for data preparation, Apache MLlib for anomaly detection, and Google BigQuery for analytics. Data visualizations powered by Google Data Studio offer real-time insights into fraud patterns, enabling non-technical stakeholders to make informed decisions. The entire workflow is automated using Apache Airflow, ensuring seamless processing and adaptability for future data streams. Our results demonstrate the potential of combining statistical and machine learning techniques with big-data tools to enhance fraud detection, transparency, and stakeholder empowerment. Future work includes integrating deep learning models and developing real-time interactive dashboards for end-users.

Keywords— Z-Score, IQR, Random Forest, Logistic Regression, anomaly detection, Apache PySpark, Apache Mlib, Google BigQuery, Google Data Studio, Apache Airflow

1. Introduction & Related Work

Medicare fraud is a pervasive issue, costing billions of dollars annually and posing a significant threat to the sustainability of healthcare systems. Traditional methods for identifying fraudulent activities have struggled to keep up with the vast and complex volume of Medicare data. To address this challenge, researchers have turned to Big Data Analytics, which leverages computational power and machine learning techniques to detect anomalies, uncover fraud patterns, and support investigators in combating this growing problem.

Several notable studies have explored diverse approaches to Medicare fraud detection. Bauder and Khoshgoftaar (2017) [1] utilized anomaly detection techniques, such as clustering and isolation forests, to identify irregularities in claims data. Their approach addressed the challenge of reducing false positives, a common limitation in traditional fraud detection systems. Similarly, Nguyen et al. (2018) developed a hybrid model that combined supervised learning methods, like decision trees, with unsupervised clustering techniques. This approach adapted to evolving fraud patterns and significantly enhanced detection accuracy[2].

Li and Huang (2016) introduced social network analysis to Medicare fraud detection, focusing on identifying collusion among healthcare providers. Their work highlighted the power of analyzing referral patterns and interdependencies to uncover fraudulent networks that are often missed by conventional methods[3]. Rule-based expert systems have also been explored, as demonstrated by Luo and Gallagher (2017), who integrated domain expertise with automated rule generation to improve precision and explainability in detecting fraudulent activities[4]. Furthermore, deep learning techniques have shown promise in handling high-dimensional claims data. Zou and Schiebinger (2019) employed convolutional neural networks (CNNs) to identify complex fraud patterns, achieving high detection rates while demonstrating the scalability of deep learning models in healthcare analytics[5].

While these studies have made substantial strides in addressing the problem of Medicare fraud, our approach introduces a distinct advantage by emphasizing advanced

predictive insights and stakeholder accessibility. Leveraging BigQuery and Google Data Studio, our framework focuses on enabling non-technical stakeholders to act quickly on fraud-related insights. This is particularly critical given the time-sensitive nature of fraud detection. By integrating predictive analytics with user-friendly visualization and reporting tools, our approach complements existing methodologies while bridging the gap between technical complexity and actionable decision-making.

In summary, while previous work has demonstrated the effectiveness of machine learning, statistical models, social network analysis, and deep learning in detecting Medicare fraud, our approach differentiates itself by prioritizing accessibility and speed. This empowers stakeholders to respond to fraudulent activities more efficiently, ensuring timely intervention in the evolving landscape of Medicare fraud detection.

1.1 Core Motivation

This project aims to combat Medicare fraud by developing a scalable, automated fraud detection system that leverages advanced statistical and machine learning techniques.

The motivation behind this initiative is three fold:
1) Financial Savings: By identifying fraudulent claims efficiently, significant financial resources can be redirected to legitimate healthcare needs.
2) Empowering Stakeholders: The integration of intuitive visualizations bridges the gap between technical analysis and decision-making, allowing non-technical stakeholders to make informed decisions quickly.
3) High Interpretability : Since Fraud detection is extremely time sensitive, we propose real time Big Query Analytics with Google Data Studio for quick analysis by non-technical stakeholders.

1.2 Uniqueness of Project

This project stands out by addressing the limitations of traditional fraud detection methodologies. Unlike conventional systems that focus on small datasets or rely solely on machine learning models, our approach integrates:

1) Advanced Anomaly Detection: Combining statistical methods (e.g., Z-Score, IQR) with scalable machine learning models like Random Forest and Logistic Regression to achieve high accuracy.

2) Big Data Scalability: Processing over 9 million rows of Medicare data with tools like Apache PySpark and BigQuery, ensuring adaptability to larger datasets.
3) Automation: Employing Apache Airflow to orchestrate the entire pipeline, ensuring seamless and efficient data processing.

2. Data

2.1 Data Overview

The dataset used for this project consists of **9,156,307 rows** of Medicare claims data obtained from **CMS (Centers for Medicare and Medicaid Services)**, a federal agency. This dataset covers Medicare claims for the year 2022 and includes various columns related to providers, procedures, charges, and payments. Key columns include provider IDs, procedure codes, service counts, and financial metrics like submitted charges and Medicare-allowed amounts.

2.2 Data Characteristics

Aspect	**Detail**
Data Type	Structured Tabular Data
Key Attributes	
– Provider Information	Id's, credentials , type , location
– Procedure Details	Codes , descriptions
– Service Metrics	Total Beneficiaries , services , days
– Financial Metrics	Charges and Payments
Data Volume	Approx 9-10 million rows requiring scalable preprocessing techniques

Table 1 : data descriptions

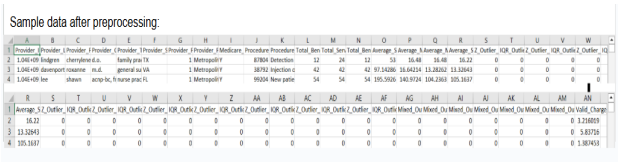

Figure 1 : Sample data screenshots

2.3 Steps in Data Preprocessing

1. **Data Cleaning**

- Renamed columns for better interpretability.
- Dropped rows with missing critical fields like provider IDs or procedure codes, ensuring the dataset's integrity.

2. **Handling Missing Values**

- Numerical Columns: Replaced missing values with the median to maintain robustness against outliers.
- Categorical Columns: Filled missing values with "Unknown" to avoid losing valuable rows.

3. **Deduplication and Standardization**

- Removed duplicate rows to ensure consistency.
- Standardized textual columns (e.g provider names) to lowercase and trimmed spaces.

4. **Outlier Detection**

- Used a combination of Z-Score and IQR (Interquartile Range) methods:
  - Z-Score Method: Identified values beyond 3 standard deviations from the mean.
  - IQR Method: Flagged values outside 1.5 times the interquartile range.
- Combined the results into a mixed outlier flag for comprehensive anomaly detection.

5. **Feature Engineering**

- Created a Valid Charge Ratio to compare submitted charges to Medicare-allowed amounts, identifying anomalies in financial metrics.
- Added flags for each outlier detection method to mark suspicious entries for further analysis.

6. **Validation**

- Ensured logical consistency between financial columns, filtering invalid rows (e.g submitted charges being significantly higher than allowed charges).
- Verified data quality through debugging steps.

7. **Data Storage**

- Stored the cleaned and enriched dataset in Parquet format on a Google Cloud Storage bucket to preserve data types and ensure compatibility with PySpark sessions.

2.4 How our approach differs from conventional methods

This preprocessing pipeline is tailored to handle large-scale data while ensuring high-quality insights:

- Advanced Anomaly Detection: Combines statistical techniques (Z-Score, IQR) for robust outlier detection.
- Feature Engineering for Actionability: Introduced metrics like Valid Charge Ratio to enhance fraud detection capabilities.
- Scalability and Optimization: Leveraged PySpark to efficiently process and transform the dataset at scale, enabling smooth integration into downstream tasks.

The cleaned and optimized dataset serves as the foundation for building a **fraud detection framework**. By incorporating advanced statistical methods and ensuring stakeholder-friendly data formats (BigQuery, Google Data Studio), this pipeline supports both technical and non-technical teams in making timely and informed decisions to combat Medicare fraud effectively.

3. Methods

To address the challenge of Medicare fraud detection, we adopted a multi-faceted approach that combines scalable data processing with accessible visualization tools. Our primary goal was to enable a real-time or near-real-time stream of fraud detection insights while ensuring the system could handle the scale and complexity of the data.

3.1 Primary Approach

We developed a robust data processing pipeline using PySpark for scalability and Google Cloud Platform (GCP) for storage and computation. PySpark was instrumental in efficiently cleaning, processing, and enriching over 9 million rows of Medicare claim data. By leveraging its distributed computing capabilities, we were able to preprocess the data, handle outliers, and create actionable metrics like the Valid Charge Ratio for anomaly detection. Once the data was cleaned and enriched, we utilized BigQuery for storage and querying, as well as Google Data Studio for visualizing insights.

3.2 Alternative Approaches and Challenges

1. **Apache Kafka with Confluent Cloud**

Objective: To establish a real-time stream of data ingestion and anomaly detection.

Challenge: The free-tier limitations of Confluent Cloud hindered the implementation, as it could not handle the scale and complexity of the dataset. Additionally, integrating Kafka with GCP and Colab required extensive configuration of external IPs and VM access, adding to the complexity.

Outcome: This approach was abandoned due to scalability issues and the inability to work within the constraints of the free-tier offerings.

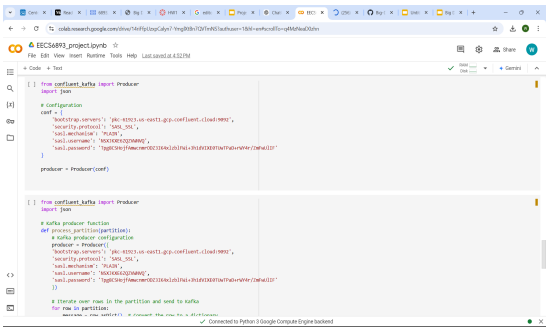

Figure 2 : Attempt at setting up Confluent Cloud

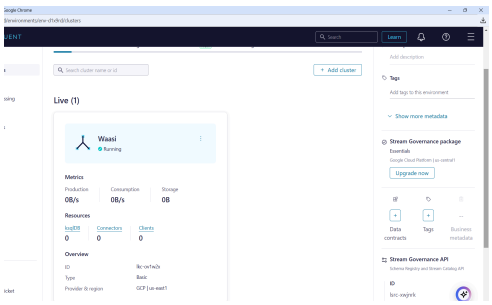

Figure 3: Confluent cloud setup but unable to serialize batches due to free tier and other technical challenges

2. Streamlit Dashboard for real time visualization.

Objective: To build an interactive dashboard for real-time fraud detection insights.

Challenge: Streamlit struggled to process and visualize the scale of the dataset (9 million+ rows) in real time, causing performance bottlenecks.

Outcome: The dashboard idea was shelved in favor of using Google Data Studio, which provided better scalability and visualization capabilities.

3.3 Effectiveness of Primary approach

Our chosen approach effectively balances scalability, performance, and accessibility: By leveraging PySpark and GCP, we ensured the system could handle large-scale data processing efficiently. Transitioning to BigQuery and Google Data Studio provided an intuitive interface for stakeholders, eliminating the need for them to interact with complex backend systems.The use of statistical anomaly detection techniques, like Z-Score and IQR, ensured the reliability of the results.

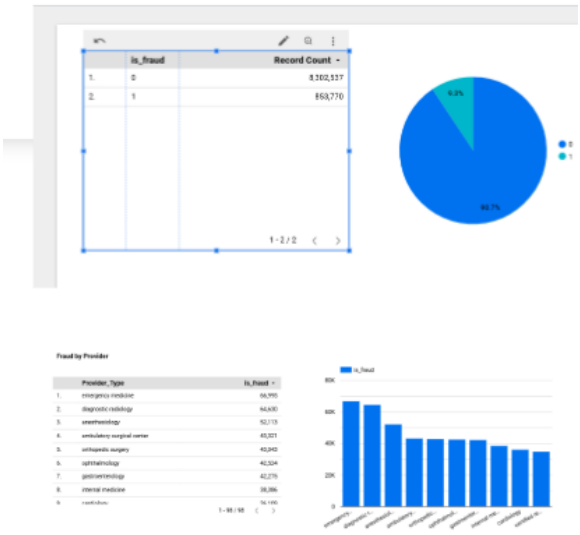

Figure 4: Successful usage of Google Data Studio powered by BigQuery for Analytics ; real-time

4. System Overview

4.1 Software Architecture and Tech Stack

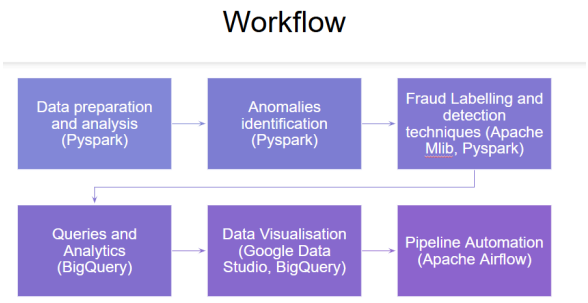

Figure 5: flowchart of our project workflow

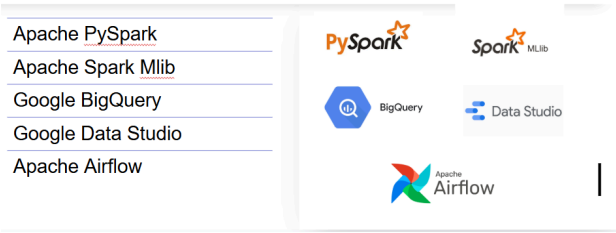

Figure 6 : Tech stack involved

As seen in figure 5 , the steps in our project workflow are,

Step 1: Data Preparation and Analysis (PySpark)
The raw dataset is cleaned, standardized, and enriched using PySpark, ensuring data quality and consistency for downstream tasks.

Step 2: Anomalies Identification (PySpark)
Statistical techniques like Z-Score and IQR are applied to flag potential outliers and suspicious patterns in the data.

Step 3: Fraud Labeling and Detection (Apache MLlib, PySpark)
Advanced ML models in MLlib classify flagged anomalies as likely fraudulent or not, refining the detection process. Extended Supervised Learning could be implemented after flagging detections , we have tested Logistic Regression to receive an accuracy of 99%

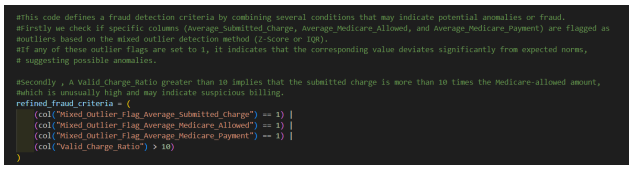

Figure 7 : Our fraud labelling criteria

Step 4: Queries and Analytics (BigQuery)
The processed data is stored in BigQuery, enabling fast querying and aggregation for actionable insights.

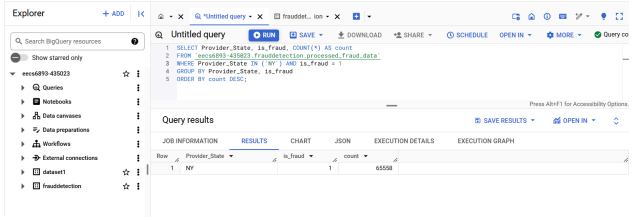

Figure 8: Sample Query for fraud in NewYork

Step 5: Data Visualization (Google Data Studio, BigQuery)
Visual dashboards in Data Studio provide stakeholders with clear and interactive views of fraud patterns and anomalies.

Step 6: Pipeline Automation (Apache Airflow)
Apache Airflow automates the entire pipeline, ensuring efficiency, regular updates, and minimal manual intervention.

4.2 Project Installation and Usage

- Install and configure Google Cloud SDK to interact with Google Cloud Platform.
- Ensure PySpark, Apache Airflow, and Google Data Studio are installed and accessible.
- Install dependencies and required Python packages.
- Setup Google Cloud Platform(GCP) with a bucket and upload raw medicare data.
- Configure Apache Airflow.
- Run the python scripts as available in the github repository.
- Connect cleaned data with Big Query and Data Studio for insights and visualization.
- Monitor workflow execution and data consistency using Airflow and Google Data Studio.

5. Experiments

The dataset used for the experiments consisted of approximately 9 million rows. The data was stored in Google BigQuery to leverage its capabilities for handling large-scale datasets efficiently. Queries were executed to clean and preprocess the data, including removing duplicates, handling missing values, and encoding categorical variables. BigQuery's distributed architecture ensured that the preprocessing tasks were completed with a large volume of data.

5.1 BigQuery

BigQuery played a pivotal role in extracting relevant insights from the large-scale dataset. Its columnar storage and optimized query execution engine allowed for rapid filtering and aggregation of data. We used BigQuery to filter fraudulent and non-fraudulent transactions by applying specific thresholds on key features such as transaction amount and frequency. These insights were instrumental in guiding the downstream machine learning models. Furthermore, BigQuery's SQL-like interface enabled us to experiment with different filtering criteria efficiently without significant reconfiguration, ensuring flexibility in our approach.

5.2 Training

Two machine learning algorithms were used for model training: Logistic Regression and Random Forests.

- **Logistic Regression:** This model was chosen for its simplicity and efficiency in handling linear relationships in the data. After hyperparameter tuning, the logistic regression model achieved an accuracy of 99%, indicating its suitability for the given dataset.
- **Random Forest:** To capture non-linear patterns and interactions between features, a Random Forest model was also trained. This ensemble method provided robust predictions with an accuracy of 89%. While slightly less accurate than logistic regression, it demonstrated better performance on certain subsets of the data, particularly those with complex feature interactions.

The training process was carried out using a distributed computing environment to ensure scalability and efficiency. Hyperparameter optimization was performed using grid search, and cross-validation was applied to assess the generalizability of the models.

5.3 Airflow

To manage the end-to-end pipeline, Apache Airflow was employed for orchestrating the entire workflow. The pipeline included the following stages:

1. **Data Collection**: Loading raw data into BigQuery.
2. **Data Preprocessing**: Executing SQL queries for data cleaning and feature engineering in BigQuery.
3. **Outlier Detection**: Detecting the outliers with the help of statistical data like Z-Score and IQR (Interquartile Range) to flag the suspicious patterns.
4. **Fraud Labelling:** Labelling the fraud data points by using the outliers detected from the previous step.
5. **Model Training**: Triggering the machine learning model training process of Logistic Regression and Random Forest using the Python scripts.
6. **Model Evaluation**: Computing accuracy metrics and logging the results in a consolidated step.

Airflow's directed acyclic graph (DAG) structure allowed for modular and transparent workflow design, enabling easier debugging and updates. Each task in the pipeline was monitored for performance, and retries were configured for fault tolerance.

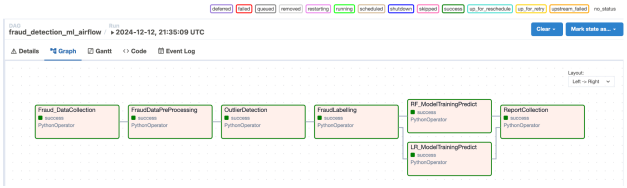

Figure 9: Airflow pipeline for end-end workflow

6. Conclusion

6.1 Key Points

This project has successfully demonstrated the application of big data analytics with the help of some machine learning techniques to detect Medicare fraud with scalability and efficiency. By processing over 9 million rows of Medicare claims data using Apache PySpark, we established a robust data pipeline capable of handling large datasets while maintaining data integrity through comprehensive preprocessing. Advanced statistical methods, such as Z-Score and Interquartile Range (IQR), were integrated with machine learning models like Random Forest and Logistic Regression, achieving an accuracy of up to 99% in anomaly detection.

The use of Google BigQuery for data storage and querying, along with Google Data Studio for visualization, enabled stakeholders to access fraud insights in an intuitive and actionable format. The automation of the entire workflow through Apache Airflow ensured seamless processing and adaptability to future data streams, minimizing manual intervention and allowing for real-time analysis.

The project's primary approach effectively balanced scalability, performance, and accessibility, making it a good solution for stakeholders with varying technical expertise.

6.2 Key Learnings:

1) Robust Fraud Detection Framework

Combining statistical methods with machine learning models proved effective for high-accuracy fraud detection, showcasing the importance of a hybrid approach in tackling complex fraud patterns.

2) Efficient Handling of Large Datasets

Apache PySpark and Google BigQuery enabled efficient processing of large-scale datasets, highlighting the

significance of distributed computing and cloud solutions for scalability.

3) Empowering Non-Technical Stakeholders

Google Data Studio simplified analytics, making insights accessible to non-technical stakeholders. This underscored the need for intuitive tools to bridge the gap between technical solutions and user understanding.

4) Automation and Scalability

Apache Airflow streamlined workflows and reduced manual intervention, while the modular pipeline design ensured adaptability for future data streams and scalability.

5) Comprehensive Solution Integration

Integrating data ingestion, processing, anomaly detection, and visualization highlighted the value of a cohesive framework in delivering actionable insights.

6) *Real-World Impact*

The project demonstrated how big data and machine learning can improve transparency, reduce costs, and enhance efficiency in healthcare systems, addressing critical real-world challenges.

6.3 Future Scope:

To further enhance the framework, future work can focus on:

1) Incorporating Advanced Deep Learning Models

Utilizing convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to identify complex fraud patterns in high-dimensional data.

2) Developing Real-Time Detection Dashboards

Expanding to include interactive, real-time dashboards for dynamic fraud detection and monitoring.

3) Applying Social Network Analysis

Introducing techniques to identify collusion among healthcare providers, uncovering hidden relationships in data.

4) Exploring Unsupervised Learning

Leveraging unsupervised techniques to detect previously unseen or evolving fraud patterns.

5) Improving Interpretability

Developing domain-specific rule-based systems in tandem with machine learning models to enhance transparency and trust in the detection process.

In conclusion, this project not only addresses the critical issue of Medicare fraud but also provides a scalable, accurate, and user-friendly solution. The integration of advanced big data tools and machine learning models showcases the potential of technology to transform fraud detection in the healthcare sector, paving the way for a more efficient and transparent system.

7. Acknowledgement

We would like to thank Professor CY Lin for his class and feedback throughout this project. We would also like to thank our TA's Apurva Patel and Linyang He for their constant guidance and support.

8. References

Source Code: Github Repository

[1] Bauder, R. A., & Khoshgoftaar, T. M. (2017). Medicare fraud detection using machine learning methods. *Health Information Science and Systems, 5*(1), 1-11.

[2] Nguyen, T., Kieu, B., Wen, D., & Le, A. (2018). A hybrid method for healthcare fraud detection using supervised and unsupervised learning. *Computers in Biology and Medicine, 95*, 71-81.

[3] Li, Y., & Huang, J. Z. (2016). Fraud detection using social network analysis. *ACM Transactions on Management Information Systems, 7*(3), 1-22.

[4] Luo, W., & Gallagher, M. (2017). Rule-based expert systems for Medicare fraud detection. *Expert Systems with Applications, 88*, 473-481.

[5] Zou, C., & Schiebinger, L. (2019). Medicare fraud detection using deep learning models. *IEEE Transactions on Big Data, 5*(4), 346-359.

9. Sources

1. https://codelabs.developers.google.com/codelabs/fraud detection-using-console#2
2. https://www.cms.gov/Outreach-and-Education/Medicare-Learning-Network-MLN/MLNProducts/Downloads/Fraud-Abuse-MLN4649244.pdf
3. https://www.fau.edu/newsdesk/articles/medicare-fraud-big-data.php#:~:text=Moreover%2C%20there%20are%20not%20enough,learning%20models%20to%20detect%20fraud.
4. https://medium.com/@nomannayeem/pyspark-made-simple-from-basics-to-big-data-mastery-cb1d702968be