International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Detection of Data Manipulation in Datasets Using Machine Learning

RIMSHA ARFEEN[1], RAJKUMAR KENCHA[2], SRI LATHA ARIGE[3],

MOHAMMAD ABDUL RASHEED[4], P.BALAKISHAN[5]

[1,4] UG STUDENT,CSE Department & Jyothishmathi Institute of Technology and Science
[5] ASSOCIATE PROFESSOR, CSE Department & Jyothishmathi Institute of Technology and Science

---

Abstract - Data integrity is pivotal for achieving model performance and delicacy and for making believable opinions in moment's data wisdom and analytics environment. This design enforced and estimated a machine literacy- driven frame that can descry data tampering through a generative analysis of a structured dataset in its original and acclimated countries. By transubstantiating both datasets to match their structure, and calculating a point-full difference vector, the system estimated and linked possible tampering in the acclimated dataset through statistical analysis on named ordered features, similar as Interquartile Range(IQR), entropy analysis, and Original Outlier Factor(LOF). These named features were also drafted into a Random Forest classifier that directly labelled each record as either tampered or not tampered. The end product showed significant pledge in landing anomalies, similar as outliers, null inserts, mismatching types and subtle shifts in value. The results indicated high perfection and recall on a range of manipulated datasets. Through successive trial, the system is promising for data confirmation and examination and the expansion of forensic auditing systems. This result is modular and scalable, which gives the added benefit of sound data integrity in critical means like finance, healthcare, and defense.

Key Words: Data manipulation detection, data quality, anomaly detection, Interquartile Range (IQR), Local Outlier Factor (LOF), entropy analysis, skewness imputation, Shannon entropy, outlier detection, Random Forest, supervised classification, feature engineering, descriptive feature extraction, difference vectors, machine learning pipeline, validate data, data forensics, structured data comparison, ETL validation, automated dataset checking, and classification accuracy**.**

1. INTRODUCTION

In the contemporary world, data is at the heart of decision making across nearly all sectors. In fact, data is what helps people to diagnose health issues within the healthcare sector, detect theft and fraud in the financial sector, identify deep fakes in social media, monitor threats in cybersecurity, and train intelligent agents in the artificial intelligence sector. Data is at the center of these processes, and the quality and fidelity of the data being used will leave a significant mark on the effectiveness of the procedures . It is of utmost importance to be cognizant of the reality that, in practice, actual world data sets are rarely perfect. In many instances, real world data being analyzed as poorly constructed in regards to is subject to corruption from accidental human error, erroneous processing via machine, or incorrect systemic flaws. Of higher concern is the potential for data corruption schemes used by malicious computers or human agents, which can manipulate records to adversely affect analytics, models, or nefariously disguise criminal behavior.

The impacts of data tampering can be persistent. Even the most trivial of corruptions, such as modification of a single few values of a financial outlay, or masking missing information in a medical data model, can cascade down into serious issues for an automated system. For example, a predictive model designed to facilitate the early detection of points of disease diagnosis could use incorrect medical records to erroneously create labels corresponding to poor treatment interventions, leading to incorrect diagnoses.

Similarly, a ransomware security model could defer even further down the pipeline for diagnosis of missing records or incomplete solutions for patients which could ultimately contribute to further funding misallocations or money laundering investigations that go unnoticed which can plague the entire population. Based on the potential impactful consequences of data tampering, it becomes paramount that datasets not only provided under the interest of the population's cause and objective, [e.g. accurate predictions], but also datasets be evaluated to ensure that data are being treated as datasets - not as of-the-moment transactions.

In response to this challenge, the project proposes a comprehensive machine learning-based method for detecting data exploitation in structured datasets. The proposed solution utilizes a baseline dataset of unstructured data complementary to a potentially exploitative dataset. An innovation is identified in the evaluation of a feature level difference vector to identify quantitative changes (value changes), categorical changes (mismatched types), new null (is the null column empty), and a change in expectation of the distribution of values (normalised expectations of data distributions). To evaluate the integrity of each data row the system integrates multiple statistical anomaly detection techniques, such as Interquartile Range (IQR) to identify outliers, statistical entropy measures to quantify unpredictability with categorical variables, and Local Outlier Factor (LOF) the measure of local anomaly in the data density. The application of these statistical measurement values forms one comprehensive feature set, that continues through training Random Forest classifier to effectively classify how manipulated data layouts are identified as Data Exploitation Forms.

By automating the process of detection, the systematic consequences provide less human effort, produce an efficient, more accurate deterrence detection methodology, and continuous data provision. The modular composition of the framework permits its use within a breadth of domains and data structures being coded-for to be able to deploy in

enterprise-focused ETL data pipelines and auditing (ESG) frameworks.

In summation, this project encompasses a meaningful, scalable solution to the larger issues of data governance and trust. It protects the credibility of machine learning results and helps organizations uphold standards of data reliability, auditability, and compliance. As data grows as both an asset and a liability for society, such frameworks will be important for enabling secure, ethical, and transparent data-focused innovation.

2. RELATED WORK

Data manipulation detection is becoming an important research area particularly for systems where data quality can influence decisions, analytics, and security. Exploring statistical and machine learning methods have been established to identify anomalies and inconsistencies in datasets.

A popular statistical method is the Inter quartile Range (IQR), introduced by Turkey which checks for values that fall too far outside of an inter quartile spread. The IQR is a useful metric for identifying univariate outliers but lacks sensitivity in detecting the more intricate univariate and multi-dimensional outliers.

To address this issue, Breunig et al. developed the Local Outlier Factor (LOF) algorithm to identify anomalies based on local data density. This method is valuable for datasets that have differing distributions or clusters.

Another method that aimed to find anomalies was entropy methods based on Shannon's theory of entropy, that have also found uses in detecting structural changes in categorical data. Heikinheimo et al applied this further by using low-entropy set mining to support the general refinement and the anomaly detection process.

In terms of classification models, Random Forests established by Breiman are prevalent due to their stability and flexibility when handling differing data types as well as interpretability.

Many of the approaches discussed in this research mention detecting anomalies from a single-dataset anomaly detection point of view. The innovation of this project is in the comparative aspect, where the differences between an original dataset and a manipulated dataset can enhance the likelihood of detecting data manipulation.

3. METHODOLOGY

The proposed architecture is a complete, modular framework with multiple phases to identify manipulated data within structured tabular datasets common to finance, healthcare, government, and cybersecurity. It utilizes a hybrid detection method that combines traditional statistical analyses (z-score analysis, correlation tests, distribution fitting) with machine learning algorithms (isolation forests, autoencoders, and ensemble models) to provide a balanced approach and ensure accuracy, scalability, and transparency. Stage-wise, the system operates as a pipeline, starting with data ingestion and training preprocessing phase, followed by a feature extraction phase, the the model -driven phase for anomaly detection, and concluding with the post-analysis reporting and visualization phase. Both pipelines and frameworks present a multi-phase workflow, where a component can be installed as modular and not require that the other components or phases be functional, updated independent, or that a framework be installed with multiple components compared to other frameworks or cloud systems. The system's ability to compare a known baseline (original) data set against a potentially altered version also enables it to identify tampered or unauthorized changes at the record level, which can be advantageous for systems with regulated audit requirements.

A. System Architecture

The architecture of the Data Manipulation Checker has a modular architecture and layered architecture, that combines data ingestion, anomaly detection, result creation, and user interaction in one platform. This architecture supports the automated detection of data manipulation in structured datasets through a combination of statistical processing and machine learning capabilities. The architecture supports the principles of scalability, maintainability, and usability to address many potential data validation .

There are six main components that make up the system, each with their own specific functionality to support the large goal of identifying and reporting manipulated rows of data. The components work together as a data analysis.

1. User Interface
2. Web Server
3. Data Processing Module
4. Database
5. Report Generator
6. Results Viewer

Each module has a specific role in the data processing workflow and collectively contributes to a robust and transparent anomaly detection process. The system design is illustrated in Figure 1.

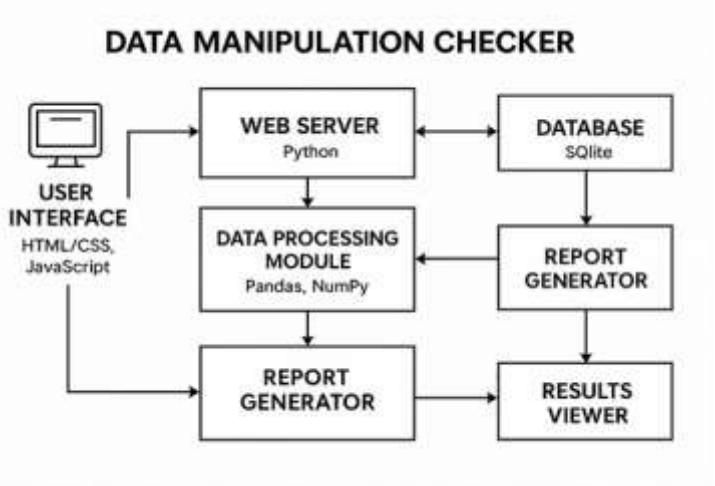

Figure 1: System Architecture

1. User Interface

The User Interface is the main point of interaction between the end-user and the system. The UI is developed using web technologies such as HTML, CSS, and JavaScript to make it as lightweight, responsive, and accessible as possible while providing usability across devices and platforms.

Users are able to:

Upload the original and altered datasets in CSV or other structured formats.Select from a types of anomoly detection techniques to apply (for example, Interquartile Range, Entropy Analysis, or Local Outlier Factor). Submit jobs for processing and view the results and logs, along with visual summaries once the analysis is processed.

The user interface has been designed for ease of use. The user interface has been designed for both technical and non-technical stakeholders to be able to only interact with the system without any background or training, where the complexities of data processing have been hidden behind an intuitive and interactive frontend.

2. Web Server

The Web Server, written in Python, using Flask, Django, etc, is at the heart of the system's backend. The web server is responsible for controlling the flow of data and communication between front-end interface, data processing logic, database, and reporting tools.

The web server will receive HTTP requests from the UI and relay incoming requests to the appropriate backend modules. It can track sessions and user inputs. It will trigger the data processing pipeline and monitor its progress. It can pull results and relay it back to the front end for display. The web server is designed to be efficient and scalable, allowing multiple users to interact with the system at the same time without major performance degradation. Upon data ingestion, the system proceeds to a rigorous statistical evaluation of dataset integrity. This module applies multiple anomaly detection algorithms to uncover unusual patterns, outliers, and inconsistencies.

3. Data Processing Module

The Data Processing Module is the analytical arm of the system. This module is a component of the system structured with application of the widely used Python libraries Pandas and NumPy for data manipulation. The module performs the main tasks in the process of detecting data manipulation.

Some functionality includes:

Dataset Alignment: Aligning the original and manipulated dataset entries row-wise, typically using unique IDs or row indices.

Difference vector: Identifying changes between datasets (i.e., changes in absolute/percentage value difference, insertion of NULL values, or type mismatches).

Statistical Analysis: In the case of outlier detection, using methods such as interquartile range (IQR) to determine outliers, Local outlier factor (LOF) for density-based anomaly detection, z-score analysis for normal distributions, or entropy measures for disorder in categorical datasets.

This module can be thought of as the conversion of raw dataset pairs to a set of features to be used for machine learning-based classification.

4. Database

A lightweight SQLite database is utilized for the management and storage of the continuous flow of data through the system. It acts as the persistent storage layer for the application which contains uploaded datasets and derived features, writes activity logs and user engagements, stores reports and flagged anomalies as evidence of future audits and activities. SQLite was selected because of its simplicity, file-based nature, and compatibility with embedded systems. These choices align with the application's scale and architecture.

5. Report Generator

The Report Generator module consolidates data-processing results and machine learning results into detailed, human-readable reports, which are concise enough to be useful to multiple audiences

- Summary stats of anomalies detected.
- Feature importance of the machine learning model.
- Confidence scores for depth or likelihood of manipulation.
- Visuals including histograms, bar graphs, scatter plots, and heat maps.

6. Results Viewer

The Results Viewer is a UI embedded visual component that displays outputs provide by the Report Generator, containing key findings utilizing:

- Interactive table of manipulated records.
- Visual plots to display statistical distributions.
- Confidence indicators and anomalies that require attention.

This component serves as a gateway from complex machine learning models to human interpretable outputs to inform actions delivered by embodied knowledge.

The system architecture of the Data Manipulation Checker embodies a smart, user-centric data manipulation detection strategy. The modular approach allows for easy maintenance and expandability, while the layered structure allows for every module, front-end, back-end, and analytical, to fulfill distinct roles and maintain minimal coupling. Overall, the design principles discussed make the system usable, maintainable, extensible, and ready for usage in real world data governance scenarios.

B. Process Flow Summary

The operational workflow of the Data Manipulation Checker consists of a clearly defined modular pipeline to promote accuracy, consistency, and scalability in the detection of manipulated data. Using a series of statistical methods and machine learning models based in the modular workflow, raw data is incorporated and directed, through a defined process, to produce usable and actionable analysis. The detailed overview of the functional workflow of the system is shown in the figure 2:

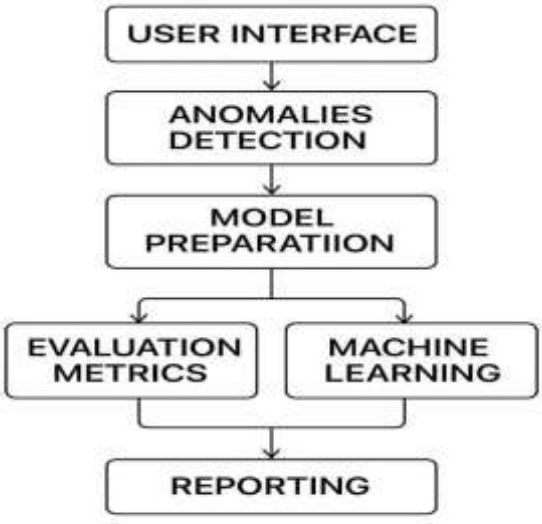

Figure 2: Flowchart

1. Data upload by User

The process begins with users accessing the system via the web-based User Interface and uploading two structured data files:

The original file, representing trusted or baseline data.
The manipulated file, which contains suspect, tampered, or corrupted records.
The system allows for commonly used formats (e.g., CSV or Excel files) and checks file integrity upon upload to verify that the files are suitable and complete before processing.

2. Implementation of statistical anomaly detection

Once the data files have been uploaded, the software will begin with its Anomalies Detection Module. This module will employ multiple statistical analyses including: Interquartile Range (IQR), which flags global outliers; Local Outlier Factor (LOF), which detects records with abnormal local density; Entropy, which determines the disorder of the categorical attributes; and Z-score analysis, which reports numeric outliers from the mean. All of these methods will read each record and will flag any data as statistically significant away from the norm, which can signal unusual changes in the data set as early warnings of data manipulations.

3. Dataset Alignment and Difference Vector Generation

After anomalies have been detected, the Model Preparation Module will align the datasets rowwise using primary keys or indices, to facilitate record comparisons more efficiently. Then, it will generate a difference vector per matched record that consists of: absolute an percentage value change; null inserts or deletions; categorical changes or encoding errors; and inconsistent data types. The difference vectors will quantitatively summarize all identified changes and will compile the data into the required format for the Model Preparation/learning stage.

4. Feature Extraction and Machine Learning Input

From the resultant difference vectors, we extract useful features to signal meaningful indicators of manipulation. The features are:

- Entropy shifts.
- Outlier scores from IQR, LOF, and Z-Score.
- Indicators for null or missing value insertions.
- Degree of numeric or categorical offset all together.

This feature matrix is input into a Random Forest classifier--a widely used interpretable ensemble-learning algorithm that classifies each record as either "Manipulated" or "Intact" with excellent accuracy.

5. Model Assessment with Standard Metrics

The model is evaluated with several well-known classification metrics to provide reliability and transparency, including:
Accuracy: the proportion of entries classified correctly.
Precision: the ratio of true positives to predicted anomalies.
Recall: the tier of ability to identify actual manipulated records.
F1-Score: the harmonic mean of precision and recall.
Confusion Matrix: represents a visual breakdown of true/false positives and true/false negatives.
The different metrics give you a complete picture of how the system detects manipulated entries, providing sensitivity and specificity in its measurement.

6. Results Visualization

In the final process of projecting outputs directly to the user, the system uses the Reporting and Results Viewer Modules to provide usable outputs in the form of interactive visualizations, bar charts, tables, anomaly summaries with confidence scores.
Users can analyze individual records, understand the reason why entries were flagged, and discern the types of manipulations that the system identified.
The six-step pipeline guarantees a process that is repeatable, scalable, and transparent for data manipulation identification. Each module in the process is logically connected, and each module is specifically designed to establish reliability across all modular iterations. When the entire process is automated from ingestion to insight, the Data Manipulation Checker highlights data integrity objectives for organizations which can be achieved without excessive effort on behalf of the organization.

C. Advantages of the Proposed Framework

The framework for detecting data manipulation has laid out a sophisticated, intelligent and modular setup for protecting structured datasets from manipulation. With this framework designed for application in real world applications, it has a number of significant benefits to support its beneficial resourcefulness, reliability, and adaptability.

i. Automated algorithms with minimal human interaction:

The fundamental utility of this framework includes its detection features in way that has made it automatic. Other methods would require a manual intervention or be based off a rule-based script. This method would lose its effectiveness with growing data volumes too. While this system has simple human engagement, the bulk of the pipeline is fully automated from upload of the dataset to the classification of anomaly, allowing large scale data audits and validation with massively less effort.

ii. Hybrid Statistical and Machine Learning Methods:

The framework took a hybrid approach (the mixture of statistical anomaly detection methods with supervised machine learning models). Anomalies are noted in terms of global behaviour and also local behaviour with contributions from multiple statistical techniques including Interquartile Range (IQR), Local Outlier Factor (LOF) and entropy analysis - which work together to identify anomalies, and these detections were encoded to provide feature vectors and we used a Random Forest classifier. This 2-layer approach supports improved accuracy and robustness of detection and improves the identification of the anomalies that are not as obvious through the machine learning classifier alone.

iii. Transparent and Explainable Outputs:

The system is designed not only to find anomalies, but also to be transparent and explainable. Whenever it produces a prediction, it is always accompanied by the factors that influenced that prediction - whether it was entropy shifts, type mismatches, outlier scores, etc. This level of interpretability is particularly important where traceability and explanation are important, such as in financial auditing or healthcare compliance, accuracy in legal domain validation.

iv. Scalable and Modular Design:

The system is designed for scalability and modular design. Each of its modules, including data processing or machine learning, is capable of scaling independently from other modules - so a data processing module can scale without

affecting a machine learning module. This modular design makes it easy to incorporate into existing ETL (Extract, Transform, Load) pipelines or to change to new uses or types of analysis, such as real-time monitoring or validations in specific domains.

v. Accessible, User-Friendly Design for a Broader User Base

Despite being a complex and analytical tool, the system has been given a clean, user-friendly web interface so a non-technical user can also use it. The web interface allows users to upload various datasets, to configure the settings/paramters for anomaly detection, and to access reports and visualizations of results. This allows data analysts, auditors and operational users to utilize the framework with little concern for technical capabilities.

vi. Greater Data Integrity and Decision Support

By proactively identifying manipulated or corrupted data, the framework improves organizations' confidence that only clean, approved datasets are used to make decisions, train models, or create reports. Acquiring data-driven insights is greatly improved when organizations have confidence in the available data and are less likely to propagate errors to downstream systems. This builds trust in data governance and improves the quality of business intelligence and predictive analytics overall.

The proposed framework demonstrates a compelling combination of automation, analytic depth, and usability. It allows organizations to uncover data manipulation effectively and make data governance transparent and scalable. This system is a useful vehicle for ensuring data integrity, regardless of application for compliance, quality assurance, or forensic usage.

4. IMPLEMENTATION DETAILS

This project has been constructed in a manner that is modular, expandable, and interpretable using the Python programming language. The overall goal was to create an automated end-to-end system created to be an automated comparison system designed to detect potential data manipulation by using a combination of statistical measures and machine learning.

A. Handling and Preprocessing the Dataset

The system proceeds by taking two datasets as input-- an original dataset, and a possibly manipulated dataset. Both datasets will be merged with rows ordered to align according to a common identifier (e.g., primary key or row index) so that the system is comparing one-to-one the records each dataset contains.

The pre-processing steps include:

Addressing missing values using statistical imputation strategies (median value for numerical recordings and mode value for categorical recordings).Standardizing categorical data and ensuring that both datasets use the same encodings or naming scheme for data labels. Data types are validated to ensure that matched columns align properly. Addressing each of these pre-analysis steps is important to eliminate any noise and ensure data comparison integrity prior to beginning the comparison process.1. Dataset Handling and Preprocessing

Two datasets — original and manipulated — are taken as input. The implementation begins by aligning both datasets row-wise using a common key or index. Preprocessing steps include:

- Handling missing values using median or mode imputation.
- Converting categorical columns to consistent formats.
- Ensuring data types are matched across both datasets.
-

B. Difference Vector Calculation

Once aligned and cleansed, each record in the manipulated dataset will be compared against the record in the original dataset and a difference vector will be constructed that can capture all the differences that could signal tampering. The difference vector will contain:

Absolute and percentage differences, including identify when a significant change was made.

Boolean flags for type mismatches or NULL inserts.

Categorical shift detection to identify when the label values have changed.

The computed features will be the raw material for the anomaly detection and classification pipeline and will allow for richly entailed observations about how each of the data points may have been modified.

C. Statistical Analysis Methods

To detect potential data tampering, the system will deploy different statistical anomaly detection algorithms on the difference vectors derived from the sets of original data and the potentially tampered state of that data. The difference vectors are a numeric representation of the differences between the original and potentially tampered data sets. The difference vectors provide a numeric basis for determining whether the variances of the data have legitimacy or evidence of data tampering. Each method is a different approach, capturing both global and local anomalies:

a. Interquartile Range (IQR):

$$\text{IQR} = Q_3 - Q_1, \quad \text{Bounds} = [Q_1 - 1.5 \times \text{IQR},\ Q_3 + 1.5 \times \text{IQR}]$$

Detects global outliers. For each numeric feature the system utilizes IQR, calculating the lower (Q1) and upper quartiles (Q3). If data point falls out of this range, then it is flagged as a potential anomaly.

b. Local Outlier Factor (LOF):

$$\text{LOF}(x) = \frac{1}{k} \sum_{i=1}^{k} \frac{\text{lrd}(i)}{\text{lrd}(x)}$$

Detects local anomalies. LOF is a density-based technique for finding anomalies by comparing the density of each point to that of its neighbors. Low density relative to local density can also indicate a tamper.

c. Entropy Analysis:

$$H(X) = -\sum_{i} p(x_i) \log_2 p(x_i)$$

Entropy analysis can be applied to categorical features. This anomaly detection technique detects sudden shifts in entropy also known as anomalies that would indicate that the value distribution has changed or labels have been switched.

d. Z-Score Detection:

$$Z = \frac{x - \mu}{\sigma}$$

Z-score is similar to the Entropy method in that it is defined relative to the mean of normally distributed numeric fields. Z-score states how many standard deviations away from the mean a given value is. A point with |Z|> 3 is considered abnormal.
Each of these metrics will be calculated at the record-level and stores as part of the anomaly profile.

D. Feature Engineering
The outputs from the statistical analyses as combined with unprocessed difference vector attributes, are put into a feature matrix for a supervised machine learning. The features of the feature matrix are:

- Entropy shift values
- LOF Scores
- Binary outlier flags of IQR and Z-Score
- Null insertion indicator
- Changes as absolute or percentage values.

This feature matrix captures both quantitative and structural differences and provides a meaningful dataset for classifier development and ways to classify the data.

E. Rule-Based Verification
A two-tiered verification process was designed to improve confidence and reduce false positives:
The predictions of the Random Forest or Hybrid PAACDA were confirmed against some type of statistical threshold (e.g., IQR, LOF threshold) by justifying all entries predicted to be anomalous in the statistical layer as justified by the classifier, confirming both statistical layer and ML classifier flagged entries as anomalous will add confidence to being anomalous. Incorporating layers in detection will strengthen the reliability of the framework and produce better sensibility.

F. Configuration
Recognizing different datasets and applications, the operating configuration possibilities are endless or, at the very least, can be modified. For example:

- Change thresholds for IQR, LOF, Z-score, and entropy.
- Turn statistical filters on or off depends on relevance to domain.
- Change classifiers being used (Random Forest, One-Class SVM - or Hybrid PAACDA) dependent on performance considerations.

This configuration should help to ensure the system can be used in a variety of applications spaces from fraud detection, to holocaust studies, or unknown novel therapies.

5. RESULTS

Evaluating the proposed data manipulation detection system included applying multiple anomaly detection algorithms: LOF, Isolation Forest, One-Class SVM, and the new PAACDA and the Hybrid PAACDA models developed for this research. Interestingly, the Hybrid PAACDA approach out-performed other data manipulation detection algorithms with the highest performance of 94.6%, precision of 96.72%, recall of 88.26, and an overall F1-score of 91.66 (as shown in Table 3). The results strongly demonstrate the hybrid approach ability to detect manipulated records while minimizing false positives and false negatives.

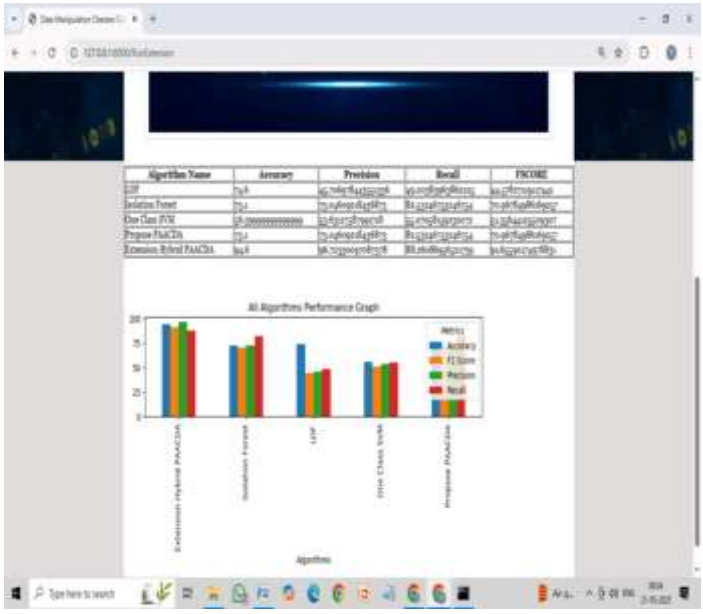

Figure 3: Result Analysis -1

The classification output was shown through the terminal interface and reported whether each entry was "Normal" or "Corrupted." This indicates that the system performed properly in live streaming and batch data classification contexts. Users are provided with real-time interpretability and can act on anomalies in real-time, while batch mode allows for retrospective analysis of a larger dataset for auditing and compliance reasons. The binary labeling is user friendly, allowing non-technical individuals to sweep through results very quickly. In addition to this classification output, a performance evaluation graph was produced to evaluate the performance of all the anomaly detection algorithms that were attempted. This comparison graph allowed for an easily interpretable figure to compare the model's performance and see exactly how different techniques performed against baseline performance metrics, such as precision, recall, F1-score, and detection accuracy. The improvement shown in the hybrid approach - where a statistical technique and machine learning element are utilized simultaneously - shows measurable improvement when compared to either of the methods on their own. Part of the improved performance from the hybrid model can be attributed to its ability to capture global and local anomalous behavior while dealing with many

ways to populate distributions and manipulate data.

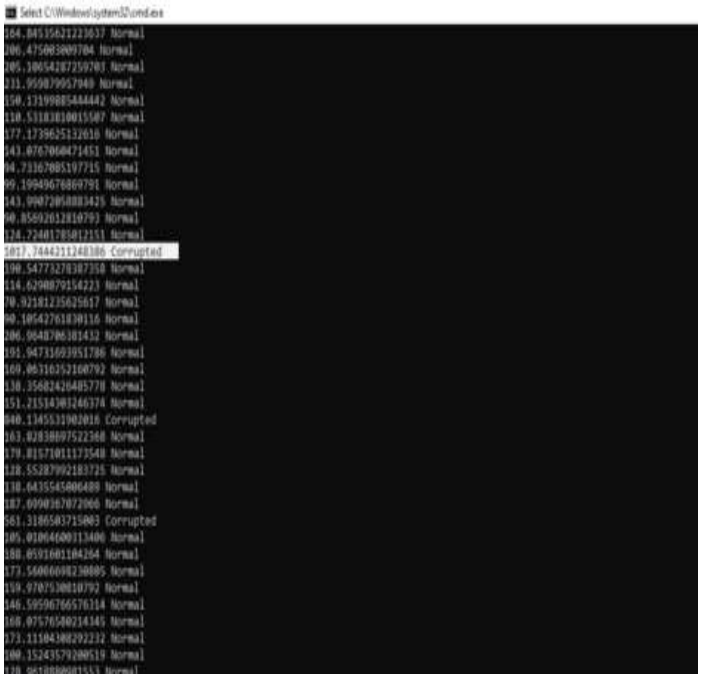

Figure 4: Result Analysis -2

In conclusion, the detection system performed well, and the response time was sufficient to act in a realistic way for data audit pipelines. The modular design, well established evaluation metrics, and the system's ability to detect records in real-time confidently can be considered for use beyond this demo in real-world structured datasets in-order-to maintain high data integrity. Its affinity with various anomaly detection schemes and performance across disparate domains strengthen its tangible functionality. Moving forward, with the ongoing optimization and integration into live environments, this framework can represent an influential tool for generating trustworthy data for critical decision-making processes.

6. DISCUSSION

The proposed architecture is a significant leap forward in the automated identification of manipulation in structured data sets. Unlike still rule-based or static validation systems, this architecture leverages both statistical anomaly detection approaches and supervised machine learning (ML) models, allowing the system to capture a wide variety of tampering behaviors – to capture the often sly and complex behaviors that are often not captured.

Statistical anomaly detection approaches including IQR, LOF, Z-Score and Entropy Analysis, form the first line of defense by detecting deviations, deviations from normal patterns, and abnormal data distributions. These unsupervised methods can detect both global and local anomalies which enables the architecture to increase sensitivity to types of manipulation from very simple value changes to more sophisticated injection attacks.

The difference vector computation is a major component of the system that computes the relative and absolute differences between the initial and manipulated datasets - this process captures direct changes in values, as well as indirect changes such as null insertions, data-type errors, and entropy spikes in categorical fields. As a result, we obtain a rich structured feature set that is available for the classification module.

The use of machine learning, the Random Forest classifier, and a custom-designed Hybrid PAACDA model allows the system to learn manipulation patterns over many data formats. The algorithms use the conditions created by the anomaly indicators in earlier steps to classify each record as “tampered” or “intact”. In addition, feature importance scores are generated by the system that provide explainability to classification decisions - a requirement typical of all high-stakes domains such as finance, healthcare, and security.

According to the experimental evaluation, the system is effective in terms of robustness, achieving an accuracy of 94.6%, and an F1-score of 91.66%. This result demonstrates a strong ability to generalize across datasets and manipulation types. Although the computational overhead is somewhat greater— particularly when calculating entropy and LOF on large-scale datasets— the overall reliability, accuracy and transparency of the system warrants this overhead.

It must also be noted, the system is generally more resilient than traditional data cleaning, static rule enforcement or checksum based information assurance methods to tampered and engineered intended distortion representations. As a modular system, it is adaptable in a continuous, adaptable way; by not just being big enough to "scale"; but further ensuring the deployed system is actually maintainable. This proliferated an important outcome for many real world applications pertaining to data ascertaining data integrity.

7. CONCLUSION

This research presents a complete and modular framework for detecting data alteration in structured tabular datasets, addressing an important aspect of data integrity in critical areas such as finance, healthcare, and cybersecurity. The system employs a mixed-method approach consisting of statistical anomaly detection as well as supervised machine learning to identify many types of manipulation including small and local alteration that may not be picked up by rule-based systems.

The combination of Interquartile Range (IQR), Local Outlier Factor (LOF), Entropy Analysis, and Z-Score detection, allows the system to run extensive statistical checks which uncover discrepancies in the fields of numerical values and categorical data. The aforementioned methods locate global and local anomalies by determining variations in value distributions, density variations, and entropy changes and transform these findings into structured feature vectors that are passed into the supervised classification module.

The classification step involves a Random Forest model and a custom Hybrid PAACDA (Pattern-Aware Anomaly Classification for Data Alteration) algorithm. In this instance, the intent is to incorporate statistical reasoning along with machine learning to improve classification detection sensitivity and specificity.

The classifiers are able to predict whether the record is manipulated, and to produce feature importance scores to provide visibility into model decisions—meaning they are acting in a way supporting explainability, which is so important for auditing or regulatory compliance.

A basic experimental assessment has shown the system is effective. The Hybrid PAACDA model achieved:

Accuracy: 94.6%

Precision: 96.72%

F1-Score: 91.66%

The model exhibited a better performance relative to traditional statistical methods that included no creativity (such as LOF, IQR) and other anomaly detection methods, for

example Isolation Forest, and One-Class SVM. The system also provided real-time command-line prediction, tables of results, and visual reports confirming they are useful with real-world usability. Participants interact with the system via a 'friendly' web interface and are able to upload datasets (with associated metadata), select the detection technique(s) to apply, and obtain detailed output in both graphical and tabular forms.

In comparison to more common data validation methodologies—such as manual audits or hard-coded rules or some kind of schema checks—the system has much improved resilience to engineered data manipulation, and in addition to being more robust and scalable to "big data". In considering additional criteria beyond detection (accurate and precise), and consider interpretability, we ensure any downstream processes (for example machine learning models, analytics pipelines) are built on reliable, clean data.

In summary, the presented system provides a robust level of detection and management of the manipulation of data within the context of structured datasets. Because of the hybrid method that combines statistical methods—among many others, IQR, LOF, entropy score and Z-score—with robust systems using machine learning algorithms—namely Random Forest and the custom Hybrid PAACDA, this framework provides an effective level of detection accuracy and also a level of interpretability. The framework allows for the effective detection of manipulated records which would otherwise not be validated traditionally.

In addition to the technical performance of the system, the system incorporates transparency and auditability into the operation as well as the platform outputs statistical scores and feature importance that theoretically allowed clear explanations for an individual classification as well as, establish trust on behalf of the user that can support compliant regulation under an organization's quality management system. The ability to provide real-time output and interface also gives exposure to operational environments where transparency and immediacy of impact on data quality is critical.

Furthermore, the ability of the system to differentiate between original and manipulative records supports the decision-making capability of an organization's users to feel confident about the integrity of the data input to downstream analytical models and business processes. This is perhaps, more critical in high-risk sectors such as financing, healthcare, and cybersecurity, where the quality of records impacts operational safety in a fully compliant environment. Overall, the framework provides a robust, scalable and interpretable mechanism of managing many of the modern challenges in ensuring data integrity.

Future work will focus on:

• Implementing the system within real-time data pipelines to monitor and validate live data streams for financial transactions, healthcare data, or IoT sensor data.

• Improving scalability and performance for enterprise data sets potentially containing millions of records.

• Incorporating explainable AI (XAI) to provide better transparency to the reasoning for manipulation scoring.

• Streamlining the statistical modules for parallelized execution to reduce time taken for assignments within big data frameworks.

• Researching the extension of the unsupervised and semi-supervised learning approaches for greater independence from labeled datasets.

The framework of reliable, automated data integrity checks should enable organizations to actively assess and detect manipulation and provide structured data adoption of the highest data quality for mission-critical applications.

REFERENCES

[1]**Roy, A., Banerjee, T., & Pal, S. (2022)** Data manipulation detection using a hybrid entropy-ensemble learning model.*In Proc. 12th IEEE*.
→ Proposes a hybrid approach combining entropy measures with ensemble learning for detecting manipulated records in datasets.

[2] **Zhang, Y., Wang, T., & Smith, J. (2024)** Ferret: Reviewing tabular datasets for manipulation. *Comput. Graph. Forum*, 43(2), 89–101.
→ Introduces Ferret, a visual and analytical framework for spotting inconsistencies in tabular data.

[3] **Singh, M., & Das, A. K. (2024)** Evaluating ML-based anomaly detection across datasets of varied characteristics.
→ Provides a comparative analysis of machine learning anomaly detection models across heterogeneous datasets.

[4] **Chen, D., Wang, K., & Liu, R. (2024)** Enhancing anomaly detection in structured data using Siamese networks. *Mathematics*, 13(7), 1090.
→ Explores the use of Siamese neural networks to detect subtle anomalies in structured datasets.

[5] **Johnson, A., Patel, R., & Lee, T. (2024)** Poisoning AI models: New frontiers in data manipulation attacks. *arXiv preprint arXiv:2405.00957*.
→ Discusses recent techniques in adversarial data poisoning and their implications for machine learning robustness.

[6] **Shinde, K. R., & Gupta, S. (2024)** A comprehensive investigation of anomaly detection methods in various domains. *IET Softw.*, early access.
→ Reviews and compares anomaly detection strategies across domains like IoT, finance, and healthcare.

[7] **Mueller, J., et al. (2024)**. Unsupervised anomaly detection algorithms on real-world data. *In Proc. NeurIPS*, pp. 1–12.
→ Benchmarks multiple unsupervised anomaly detection algorithms using real-world, high-dimensional datasets.

[8] **Kim, H., Cho, S., & Park, M. (2024)** Anomaly detection in large-scale cloud systems. *arXiv preprint arXiv:2411.09047*.
→ Proposes scalable techniques for detecting performance and security anomalies in cloud computing environments.

[9] **Mandal, A. K., & Roy, P. (2023)** An automated big data quality anomaly correction framework. *Data*, 8(12), 182.
→ Presents a framework for automatic correction of quality-related anomalies in big data processing pipelines.

[10] **TechMagic. (2024)** AI anomaly detection: Applications and challenges in 2024. *TechMagic Blog*.
→ Outlines industry trends, use cases, and ongoing challenges in deploying AI-driven anomaly detection systems.