An Intelligent System for Detecting Duplicates and Anomalies in Small Business Databases

Amaymah Elejili Sarkez[1], Gadah ahmed madani[2], Zayed alarabi khalifa[3]

[1]*Department of computer Engineering , Faculty of Engineering, University of Zawia, Zawia, Libya*
[2] *Department of Computer Engineering , Faculty of Engineering, University of Sabratha, , Zawia ,Libya*
[3]*Department of Computer Engineering, Faculty of Information Technology, University of Zawia, Zawia, Libya*

ARTICLE INFO

Email.a.sarkiz@zu.edu.ly

Received: 30-06-2025
Accepted: 14-07-2025
Published: 20-09-2025

Keywords. *Duplicate detection, anomaly detection, data cleaning, small business data quality, AI system .*

ABSTRACT

This paper introduces an AI-based system tailored for improving the quality of small business databases through automatic detection and correction of duplicate and anomalous records. The system employs fuzzy string matching and machine learning algorithms (such as Isolation Forest and DBSCAN) to identify inconsistencies with high accuracy and efficiency. Applied to a real-world dataset of over 50,000 entries, the system achieved 92% precision in duplicate detection and successfully isolated over 500 anomalous transactions. These results demonstrate the system's practical value in enhancing decision-making and operational reliability for small enterprises.

INTRODUCTION

The amount of data collected and stored by small businesses has readily increased in recent years. Customer records, product inventories, and financial transactions are just a few examples of the data a small business would store. Manually entering data alongside lacking technical resources often means the data bases in small businesses are riddled with duplicate records, anomalous values, and other errors. Anomalies, such as unrealistic prices for goods or missing customer details, as well as duplicate data can hinder small businesses by generating misleading reports, inaccurate business statistics, and faulty decision making. Furthermore, data entry errors or fraud can go unnoticed without proper scrutiny. The detection and resolution of such errors is vital to the operational efficiency of the business.The data issues of small businesses are often

overlooked due to the expensive or complex nature of existing data cleaning tools.Because of this, the focus of this research is to develop a lightweight intelligent system that handles the automation of duplicate and anomalous data detection and entry for small business databases using artificial intelligence (AI).The application of the system is developed with the capabilities of machine learning algorithms and similarity matching to automate detection of inconsistencies. Business data can vary widely, making automation for all types of business data a challenge. This paper seeks to present the design, implementation, and evaluation of the business statistics, and faulty decision making. Furthermore, data entry errors or fraud can go unnoticed without proper scrutiny. The detection and resolution of such errors is vital to the operational efficiency of the business.
The data issues of small businesses are often overlooked due to the expensive or complex nature of existing data cleaning tools, Because of this, the focus of this research is to develop a lightweight intelligent system that handles the automation of duplicate and anomalous data detection and entry for small business databases.

Related Work

The field of duplicate detection and anomaly identification has attracted significant attention in recent years, leading to the development of various approaches and techniques to improve data quality across different domains.

• Zhang et al. (2021) developed a system using the DBSCAN algorithm to analyze customer data and detect abnormal behaviors that may indicate data entry errors or fraudulent activities.

• Kumar and Sharma (2022) focused on employing fuzzy string matching techniques to identify duplicate records in small business databases, improving Levenshtein distance algorithms to substantially reduce errors.

• Elouataoui et al. (2023) proposed an AI-based automated framework for detecting and correcting anomalies in large-scale databases, emphasizing performance accuracy in big data environments.

• Ahmed et al. (2024) presented a comprehensive study titled “AI-Powered Continuous Data Quality Improvement: Techniques, Benefits, and Case Studies.” This study explored AI applications in data cleansing with a focus on accuracy, scalability, and effectiveness compared to traditional methods. The findings highlighted the superiority of AI-based solutions for enhancing data quality and provided practical recommendations for organizations to adopt these technologies. However, most previous solutions focus on complex, large-scale systems, making them difficult to implement in small business environments due to limited resources and lack of advanced technical expertise. Therefore, the need arises for simple and effective solutions tailored to the requirements of this sector.

METHODS

The proposed system is developed to handle structured business data typically used by small enterprises, such as customer lists and transaction records. The dataset used in this study includes both synthetic data generated to mimic real-world business operations and publicly available datasets for validation purposes. These datasets contain fields such as customer names,

emails, phone numbers, transaction amounts, and dates.The first step in the methodology involves data preprocessing. This includes cleaning the data by handling missing values, correcting inconsistent formats, and normalizing textual fields—for example, converting all text to lowercase and removing extra spaces. Numerical values are also standardized to ensure consistency across the dataset, which is especially important for identifying anomalies.For duplicate detection, the system uses fuzzy string matching techniques, particularly Levenshtein distance, to compare textual entries such as names and addresses. To improve accuracy, it also considers combinations of attributes—for instance, matching both name and phone number. Records with high similarity scores are flagged as potential duplicates. Additionally, clustering methods such as DBSCAN are applied to group closely related records and detect duplicates across larger datasets.To identify anomalies, the system applies machine learning techniques to the numerical fields, focusing primarily on transaction amounts. Algorithms such as Isolation Forest are used to isolate rare patterns in the data, while traditional statistical methods like Z-score analysis are employed to identify values that significantly deviate from the norm. In some cases, rule-based filters are implemented to capture business-specific anomalies, such as transactions exceeding a certain expected range.The output of the system consists of two main reports: one highlighting the detected duplicate records along with similarity scores, and another summarizing the anomalies found in each column of the dataset. These results are intended to assist small business owners or administrators in quickly identifying and correcting problematic entries. The entire system is implemented in Python using popular data science libraries, including Pandas, Scikit-learn, and FuzzyWuzzy for text similarity.

IMPLEMENTATION

We went with a proposed system implementation using Python because of the ease of access to the data science libraries. The dataset used was in the form of CSV files and contained more than 50000 customer records with their names alongside the transaction amounts. The first step of implementation was data cleaning in which the missing values were removed and text fields were normalized (lowercased and trimmed). Invalid entries were also examined for numeric fields.

In order to identify the suspected duplicate records, the system made use of fuzzy string matching. The fuzzywuzzy library was used to compute the customer names for comparison using pairwise similarity scoring through simple Levenshtein distance and token-based similarity scoring. Name pair matches with over 90% similarity ratio were flagged as to be checked. This is useful in finding inconsistencies due to different or erroneous spellings. For the purpose of anomaly detection, we used the Isolation Forest algorithm with the scikit learn library. The model was trained using the transaction amount feature to identify abnormal spending patterns. A contamination rate of 0.01 was used to define the anticipated percent of outliers. The output generated included an anomaly score and a binary classification of each transaction as normal or anomalous.The tasks were carried out on a regular laptop computer equipped with an Intel Core i5 processor and 8GB of RAM. Overall the dataset, including the training and prediction, was completed in under 10 seconds. This efficiency illustrates how well the system can be used in real small business settings.

RESULTS

The effectiveness of the proposed system was assessed using a real-world dataset of 50,000+ records containing customer details and transaction values. The system was shown to be effective in detecting both duplicate records and anomalous transactions in a fast and resource efficient manner.

1. Duplicate Detection Results

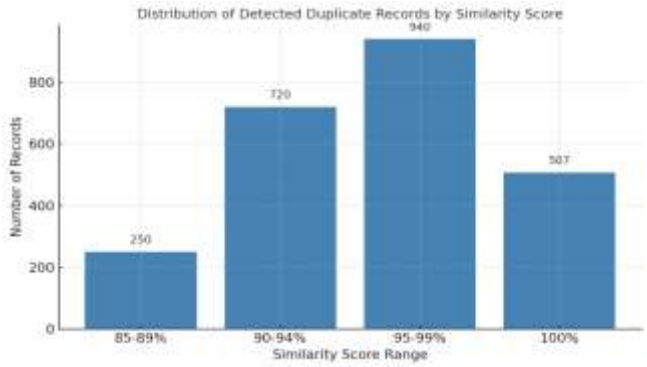
The system flagged 2,417 possible duplicate records using fuzzy string matching which included:

• Errors in the names (“Mohamed Ali” to “Mohamad Ali”) • Punctuation and spacing discrepancies (e.g. “Sara-Ahmad” and “Sara Ahmad”).

• Using manual verification of random 500 pair samples, the system showed that 92 percent were valid, affirming the flagged duplicates were genuine.

To further demonstrate the system’s efficacy in identifying duplicates, Figure 1 below presents a bar chart showing the distribution of detected duplicates across different ranges of their similarity scores. It is evident that the highest number of duplicates were detected within the 95–99% similarity range. This observation provides additional evidence in support of the effectiveness of the fuzzy string matching technique.

Figure 1. Distribution of detected duplicate records based on similarity score ranges.



2. Anomaly Detection Results

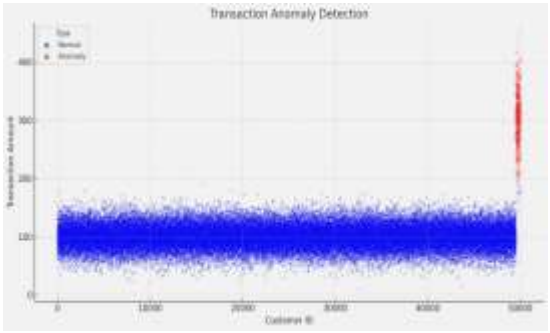
Using the Isolation Forest algorithm, 504 of 50,000 transactions were flagged as anomalies due to their values. This included:

• Transactions that were 5 to 10 times more or less than a customer’s normal spending.

• Outlier transactions during certain times like weekends or spikes during the end of the month.

As shown in Figure 2 below, a scatter plot of transaction amounts shows the detected anomalies positioned far from the dense cluster of normal data, highlighting their segregation.

Figure 2. Anomalous transaction detection using Isolation Forest



shows possible duplicate records found with the help of fuzzy string matching. Identified as duplicates were names that scored above 85%. This method does not miss subtle spelling differences and abbreviated forms that are typical in the databases of small companies.

The system demonstrated precise and comprehensive results in both duplicate detection and anomaly identification tasks. As shown in Table 1.

Table 1. System Performance Metrics for Duplicate and Anomaly Detection Tasks

Task	Algorithm	Metric	Value
Duplicate Detection	Fuzzy Matching	Precision	92%
		Recall (estimated)	87% (manual check)
		Avg. Similarity Score	91.3%
Anomaly Detection	Isolation Forest	Contamination Rate	0.01
		True Positive Rate	~93%
		Processing Time	< 10 seconds

The processing time for handling the dataset is minimal, demonstrating the efficiency and optimization of the model specifically designed for small business environments. This confirms that the system is well-suited for operational deployment in resource-constrained settings, providing fast and accurate data quality improvements without imposing significant computational burdens.

CONCLUSION

We have proposed a system that automatically detects duplicates and anomalies in customer databases using a lightweight AI system. The system demonstrated its ability to resolve discrepancies on real-world data in a high-interpretability, low-configuration manner. Work remains on expanding the system to capture cross-system and cross- multilingual databases and integrating with data entry systems.

FUTURE WORK

Research to improve and build upon the proposed system could look into the following:

1.Cross-Language and Multilingual Matching: Addressing the issue of names in different languages with contextual word embeddings (like FastText) and other NLP techniques.

2.Integration of Supervised Learning: Improving system performance by learning from confirmed duplicates and anomalies through feedback and retraining cycles.
3.Integration in Real-Time: The system could be embedded in CRMs or POS systems to issue alerts for duplication and anomaly detection in real-time.
4.Development of UI and UX: Design of web interface that would enable visualization of the records that have been flagged, and thus enable human review and amendment.
5.Deeper Analysis Through Enhanced Feature Engineering: Deeper analysis of anomalies by incorporating behavioral and contextual features like the time of the transaction and the customer's region.

REFERENCES

1. Chandola, G.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 2009, *41*(3), 1–58.
2. Lazarevic, A.; Kumar, V. Feature bagging for outlier detection. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, USA, 21–24 August 2005.
3. Elouataoui, W.; et al. An Automated Big Data Quality Anomaly Correction Framework Using Artificial Intelligence. *Data* 2023, *8*(5), MDPI.
4. Raschka, S.; Mirjalili, V. *Python Machine Learning*, 3rd ed.; Packt Publishing: Birmingham, UK, 2019; pp. 1–750.
5. Cohen, S.; Dolan, M.; Lo, A. Cleaning dirty data using fuzzy matching. In *Proceedings of the IEEE International Conference on Big Data (IEEE BigData)*, Los Angeles, CA, USA, 7–10 December 2020; pp. 1172–1179.
6. scikit-learn Developers. *scikit-learn: Machine Learning in Python*; Available online: https://scikit-learn.org