Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 236-249
Doi: 10.18466/cbayarfbe.1677875
Research Article

Celal Bayar University Journal of Science

Improving the Quality of Enterprise Data Management with Tree-Based Models

Erdal Büyükbıçakcı[1*]

[1*] Sakarya University of Applied Sciences, Information Technologies Vocational School, Department of Computer Technologies, 54500 Sakarya, Türkiye.
* erdal@subu.edu.tr
* Orcid No: 0000-0001-7276-741X

Received: April 16, 2025
Accepted: February 16, 2026
DOI: 10.18466/cbayarfbe.1677875

Abstract

Data credibility is essential for reliable decision-making and decision-support systems across Salesforce environments during the processing of transactional data as observed in the Online Retail Dataset. This work analyses the application of tree-based machine-learning models in improving data quality through transformation and cleaning processes for the removal of missing values, duplication, and outliers. The approach includes data preparation, relevant feature selection, model construction, and deployment within Salesforce processes to monitor data quality in real time and batch workflows. With tree-based models, substantial performance gains were observed, with precision up to 90.8%, recall up to 74%, and overall accuracy up to 91.6%. After Salesforce integration, completeness increased by 12%, accuracy by 10%, and consistency by 15%. The system's retraining mechanism and feedback loop ensure protection against long-term data degradation in enterprise CRM environments.

Keywords: Artificial intelligence, Customer relationship management, Data quality improvement, Tree-based models, Machine learning, Salesforce.

1. Introduction

Data quality remains key to business operations and analysis, especially for Customer Relationship Management (CRM) such as Salesforce [1]. High-quality data form the foundation of reliable analysis and managerial decision-making. However, like many other business databases, CRM systems often suffer from certain fundamental data quality issues such as missing fields, data duplication, outliers and data inconsistency which can reduce the functionality as well as reliability of the system significantly [1-2]. As organizations work with large volumes of data, the use of automated and scalable methods for increasing data quality has become a compelling issue. Machine learning (ML) is one technology that appears to hold great potential for data-quality improvement [3-4]. Rule-based systems [5] struggle with the inability to learn more patterns of anomalies especially when they are increasingly sophisticated while ML techniques [6] such as tree-based ensembles Random Forest (RF), XGBoost (XGB) and LightGBM (LGBM) are well suited to modeling complex relationships within the data and also identifying errors, tree-based models such as RF, XGB, and LGBM allow the ensemble system to learn where new models are trained to fix the shortcomings of previous models giving it high predictive power [7-8].

As is evidenced by the substantial body of research which has been published on the subject, machine learning (ML) techniques, including ensemble methods such as tree-based ensembles (RF, XGB, LGBM), have been shown to be an effective means of improving data quality [9]. However, a discernible gap persists in the literature concerning end-to-end, deployable frameworks tailored for live enterprise CRM environments such as Salesforce. A plethora of studies have previously concentrated on the algorithmic performance of XGBoost or LightGBM in offline contexts or for general anomaly detection, without demonstrating a holistic pipeline that integrates domain-specific feature engineering, real-time/batch processing, and a continuous improvement feedback loop within the specific constraints and workflows of a major CRM platform.

The present study addresses this lacuna by transcending the confines of a purely algorithmic comparison. The primary contribution of this work is the design, implementation, and validation of a comprehensive framework that operationalizes tree-based models (with

LightGBM as the deployed model) for data quality management directly within Salesforce. This distinguishes the present study from previous work in the field. The methodology presented herein is intended to be systematic and the methodology is designed to be applicable to both transactional retail data and model deployment. Furthermore, a practical pathway is proposed for both real-time anomaly correction and batch processing, and it is claimed that this provides a replicable solution for enterprise data management.

The objective of this study is to demonstrate and quantify the effectiveness of the integrated tree-based framework (RF, XGB, LGBM), with LightGBM as the final model framework in improving data quality within Salesforce systems. The Online Retail Dataset was used as a representative case to illustrate this effect [10]. The primary steps in this process are data preparation, feature engineering, model training, and subsequent deployment for real-time and batch processing. The performance of the model is evaluated through precision, recall, F1-score, and observable improvements in key data quality dimensions.

The results of this research show that tree-based models improve data quality to a greater extent compared to traditional methods [11]. Therefore, due to the scalability and the flexibility of the proposed model, this work makes a contribution towards providing a solution on how to handle data quality problems within large–scale, complex enterprise CRM systems [12]. This work adds to the literature in determining the effectiveness of machine learning in data quality management and sets the background for the research in this field.

The present study analyses the application of tree-based models in improving data quality through transformation and cleaning processes for the removal of missing values, duplication and outliers. The approach encompasses the preparation of data, the selection of relevant features and the construction of multiple learning models, followed by deployment within Salesforce processes to monitor data quality in real-time and in batch workflows. With tree-based models, substantial improvements were observed, with precision up to 90.8%, recall up to 74%, and overall accuracy up to 91.6%. Following the integration of Salesforce, there was an increase in data completeness of 12%, an improvement in accuracy of 10%, and a rise in consistency of 15%. These results demonstrate that measurable operational benefits can be achieved beyond the scope of traditional machine learning approaches [12].

In this paper, Section 1 presents the introduction. Section 2 reviews the related literature. Section 3 describes the methodology. Section 4 presents the results and discussion. Finally, Section 5 concludes the study.

2. Related Work

This literature review brings together current and previous studies and suggests that tree-based ensembles (RF, XGB, LGBM) could be a scalable solution for a large-scale data management system such as Salesforce. It also reinforces the need to continuously monitor and update features related to data quality.

Data quality in CRM systems has been of significant interest because it directly affects the reliability of analytics and decision outcomes. Chakraborty et al. (2024) [13] assert that poor data quality results in major operational costs and losses. Recent work by Suh (2023) [14] underscores that data completeness, consistency, and accuracy are essential ingredients required for sustaining and growing CRM. Salesforce is one of the most common CRM platforms and requires robust measures aligned with these dimensions, as suggested by Shahbaz et al. (2021) [15].

Examples of these shortcomings include the limitations of rule-based systems; the traditional methods (Kedi, et al. (2024) [16]) of data quality management are inadequate to address many of the complexities associated with data anomalies. The authors stated that such systems are inadequate in handling highly dynamic and large data sets. Furthermore, static rules are ineffective in capturing emerging error patterns an aspect seen commonly in transactional datasets, demanding adaptive and automated systems (Mollá, N., et al. (2022) [17], Lim, S., et al. (2021) [18]).

Whilst traditional rule-based systems are ineffective in capturing the evolving nature of data anomalies in dynamic transactional datasets (Mollá et al., 2022 [17]; Lim et al., 2021 [18]), advanced algorithms offer more adaptive and flexible solutions to capture complex patterns (Liso et al., 2024 [19]). Among these, tree-based ensembles, particularly Gradient-Boosting-based implementations (XGBoost, LightGBM), have demonstrated efficacy and stability (Bentéjac et al., 2021 [20]). In contrast to single complex models, gradient–boosted trees build robustness by sequentially correcting errors of simpler predictors. This iterative process renders it exceptionally well-suited for data quality management, as it is inherently designed to identify and focus on difficult-to-classify instances, which often correspond to data anomalies found in noisy and imbalanced datasets. As demonstrated by XGBoost (Nassif et al., 2021 [21]) and LightGBM (Siddique et al., 2017 [22]), powerful implementations have the capacity to enhance this capability with high computational efficiency. This renders them viable for processing the substantial datasets that are characteristic of enterprise CRM systems, and effective at correcting data artefacts (Zhao et al., 2024 [23]). To provide a clearer overview of these foundational approaches, a comparative summary is presented in Table 1.

Table 1. Comparative Summary of Key Studies in Data Quality Management.

Study	Method(s) Used	Dataset	Key Findings / Contribution
Kedi et al. (2024)	Traditional / Rule-Based	SME Social Media	Inadequate for dynamic and large datasets.
Bentéjac et al. (2021)	Comparative Analysis (incl. GB)	General Benchmarks	Gradient Boosting demonstrates high accuracy and stability.
Liso et al. (2024)	Deep Learning Review	Industry 4.0	Provides a broad review of ML/DL methods for anomaly detection.
Nassif et al. (2021)	XGBoost	General Anomaly Detection	Showcased the specific application of XGBoost for anomaly detection.
Siddique et al. (2017)	LightGBM	Massive Network Datasets	Highlighted LightGBM's speed and scalability for large-scale data.
This Study	Tree-Based Models (RF, XGBoost, LightGBM)	Salesforce CRM (Online Retail)	Presents an end-to-end deployable framework for real-time and batch data quality correction in an enterprise system.

Feature engineering remains critically essential to improving the performance of machine learning models for data quality work (Verdonck, T., et al. (2024) [24]). It has been pointed out by the author Wang, J. F. (2023) [25] that greater emphasis should be placed on deriving the right features in order to define the patterns of anomalies well enough. With regard to CRM data characteristics, derivative factors such as average transaction rate, and total monetary value have emerged as informative pointers towards data quality adversity as pointed by Ledro et al. (2022). [26].

Data preprocessing plays a significant role in enhancing performance of the model in an anomaly detection problem Larriva-Novo et al. (2020) [27]. Imputation techniques and outlier removal and the effects they have on the interpretations made in a downstream analysis are also highlighted by them. In the present case, the given methods are applicable to transactional datasets, as they can involve missing and inaccurate entries. The real-time detection of anomalies is necessary in providing the quality check regarding data in the changing environment of CRM solutions [28]. Other reviews by the authors Ledro et al. (2023) [29] have emphasised the need to incorporate ML models into the data pipeline so as to facilitate the mechanical and timely correction procedures. Mathematical modeling ensures data integrity in decision-making [30], enabling Salesforce workflows to maintain constant checks.

ML model performance significantly influences data quality assessment tasks; therefore, it is imperative to use appropriate and up-to-date evaluation criteria [31]. As the authors pointed out precision, recall and F1-score are considered when evaluating classification models specifically in contexts of imbalanced data [32]. These metrics give a general picture of the performance of any model, and also give the idea as to the trade-off between false positives and false negatives for eliminating false negatives or vice versa.

It is important to note that feedback loops of the model must be performed on a continual basis in order to sustain the model's effectiveness. The authors also emphasize the idea of retraining the models with corrected predictions to adapt to shifting data patterns. This approach helps the model remain relevant in a dynamic CRM environment [33-34].

Tree-based models have been successfully applied in several CRM analytics tasks, including customer classification and churn forecasting [35]. Prior studies demonstrate their capability in modelling complex b behavioral and transactional patterns, making them suitable candidates for anomaly detection and data quality correction. Their use within Salesforce environments is therefore consistent with existing CRM machine learning research [36-37].

While the extant literature establishes the potential of tree-based ensembles, a direct, statistically robust comparison of their performance for data quality within a specific CRM context, followed by the deployment of the superior model, remains a contribution this study aims to make. The subsequent section will thus delineate the methodology that has been employed in order to facilitate a comparison between these models and to validate the proposed framework.

3. Materials and Methodology

A systematic approach supports the effective use of employing tree-based models in enhancing data quality in Salesforce. Consequently, the proposed methodology concentrates on improving data quality in Salesforce by using tree-based models, which are stable machine-learning methods, using the Online Retail Dataset. In the case of tree-based boosting learners (such as LightGBM or XGBoost), the model builds an ensemble of shallow decision trees sequentially, where each tree is trained to correct the residual errors of the previous one. This residual-correction mechanism enables the model to

focus on hard-to-classify records, which aligns well with CRM-level data anomalies such as missing CustomerIDs, duplicated invoices, and negative quantities. This approach is suitable for addressing data-quality issues in the dataset, including missing, duplicated, and inconsistent values. Using this back-and-forth loop of feedback processing, the model improves its ability to recognize anomaly and correction patterns. The systematic methodology employed in this study is visually detailed in Figure 1. The framework comprises two pipelines: (i) an offline model-training pipeline (from data extraction to the final LightGBM model selected among tree-based candidates), and (ii) a deployment pipeline for Salesforce, featuring an Apex Trigger for real-time validation and a MuleSoft/ETL batch job for periodic large-scale checks. The deployment pipeline includes two discrete integration patterns: firstly, a real-time anomaly detection workflow initiated by an Apex Trigger for immediate record validation; and secondly, a batch processing workflow orchestrated by a MuleSoft/ETL process for periodic, large-scale data quality assessments. This dual approach ensures both immediate and comprehensive data quality monitoring.

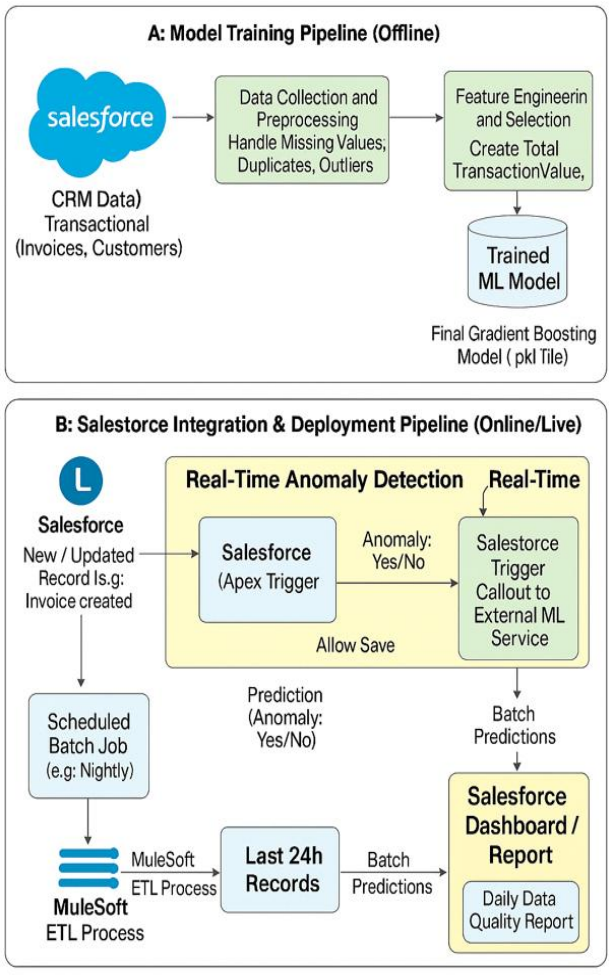

Figure 1. Technical workflow of the end-to-end data quality framework.

The goal of this stage is to incorporate a tree-based model (LightGBM as deployed) into the Salesforce platform for monitoring the quality of data in real time. Using relevant fields, including customer IDs, invoice and transaction details in the retail dataset, the model infers error-prone sections and offers corrections for accurate Salesforce reports and analytics.

3.1. Data Collection and Preprocessing

Generally, Salesforce data can contain both structured and unstructured fields, hence the need for a data preprocessing strategy. Multiple pre-processing techniques were used to develop high-quality data for the tree-based model family (RF, XGBoost, LightGBM), with LightGBM used in deployment. The Missing CustomerID values were handled using pattern-based imputation and missing Description values were filled through StockCode mapping with additional TF-IDF similarity-based imputation. A Z-score based approach detected outliers in Quantity and UnitPrice while duplicate records were eliminated in the same step. The preprocessing strategy introduced three feature groups: TotalPrice, PurchaseFrequency, and time-based attributes (e.g., month/day/hour), alongside one-hot encoding for categorical fields (Country, StockCode). Min-Max scaling was applied to normalize numerical attributes, preparing a dataset that achieved both efficient anomaly detection and correction capabilities in Salesforce.

3.1.1. Dataset Description and Summary Statistics

The study relies on the Online Retail Dataset that includes transactional records of purchases from UK-based customers of an online retailer. It covers all transactions occurring between 01/12/2010 and 09/12/2011 and comprises 541,909 instances. The dataset establishes various attributes which provide necessary information regarding individual transactions. Table 2 summarizes its attributes.

3.1.2. Splitting Strategy

The dataset contains 541,909 transactions recorded between 01/12/2010 and 09/12/2011 for a UK-based non-store online retailer (Online Retail dataset). The dataset includes 4,372 distinct customers and transactions from multiple countries. For model development and evaluation, the data were split into training, validation, and testing subsets as follows:

- 70% (379,336 records) → Training set
- 15% (81,286 records) → Validation set
- 15% (81,287 records) → Testing set

The following Table 3 summarizes numerical attribute characteristics in the dataset.

Table 2. Dataset features description.

Feature Name	Description	Data Type
InvoiceNo	Unique invoice number for each transaction	String (Categorical)
StockCode	Unique product identifier	String (Categorical)
Description	Name of the product	String (Categorical)
Quantity	Number of items purchased in the transaction	Integer
InvoiceDate	Date and time of the transaction	DateTime
UnitPrice	Price per unit of the product (in GBP)	Float
CustomerID	Unique identifier for the customer	Integer (Categorical)
Country	Country where the customer is located	String (Categorical)

Table 3. Summary of dataset statistics.

Feature	Mean	Median	Min	Max	Std Dev
Quantity	12.06	3	-80995	80995	218.08
UnitPrice	4.61	2.08	0.00	38970	96.76

3.1.3. Data Imputation Technique

Several data-quality improvement approaches were applied to the Online Retail Dataset by handling missing and incorrect data points. The process of imputing CustomerID gaps used pattern-based methods to select the most common CustomerID that occurred in each invoice. When no relationship between variables existed, mode imputation was applied; otherwise, Unknown_Customer was used as a placeholder. The Description column received its missing values from two methods: StockCode mapping provided the most common description for analogous products and TF-IDF similarity- nearest-neighbor imputation was applied to uncommon products.

The underlying rationale for these choices was domain-specific. With regard to CustomerID, a pattern-based approach was selected for the purpose of imputation, whereby the most common ID within an invoice was assigned. This approach was chosen in preference to simple mode imputation, on the basis of the logical premise that all items in a single transaction should belong to the same customer. In a similar manner, for the Description field, a TF-IDF similarity approach was employed for rare products, as it can infer semantically related descriptions, which is more robust than using a generic placeholder.

Several imputation strategies were used to handle incorrect UnitPrice data points by applying stock-code-specific median values along with regression predictions to complete missing or zero entries. The model corrected negative Quantity values using multiple methods that included return identification plus absolute value correction along with outlier removal. The use of these imputation techniques enhanced the dataset quality so it became suitable input for the tree-based models, enabling anomaly detection and subsequent remediation; LightGBM was adopted for deployment.

3.2. Feature Engineering and Selection

Feature engineering directly affects the performance of tree-based models, especially LightGBM. In this case, domain knowledge is used to extract features such as metrics on customer behavior, interaction frequency, or even signs of anomalies from Salesforce data. Statistical tools such as Principal Component Analysis (PCA) or mutual information are used to select the most informative features for the model so as to enhance its performance by lowering its dimensionality. Particular attention was given to meaningful structures and relationships typical of Salesforce datasets.

Feature engineering creates informative variables that help the model discover issues with data quality. New variables including TotalTransactionValue, derived from Quantity and UnitPrice; the number of transactions and the average transaction size per customer were created. Other novel attributes such as a customer's transaction day and time are included to capture customer purchasing patterns. The impetus behind the creation of these aggregate features (e.g., TotalTransactionValue, PurchaseFrequency) was to transform static transactional data into behavioral indicators. It is hypothesised that these engineered features function as more effective predictors of data quality issues, given that anomalies are frequently more apparent in aggregated customer behavior patterns than in individual raw data points.

Mutual information scores or RFE are used in the feature selection process to retain the most relevant features. Additional features are selected based on domain-related knowledge, such as identifying the most frequently purchased products or the most valuable customers. This step ensures that the proposed model considers important aspects of data quality related to the retail dataset and operational Salesforce business workflows.

3.2.1. Newly Engineered Features and Their Formulas

The predictive capability of the tree-based models was enhanced through the creation of newly engineered features extracted from existing dataset attributes. As illustrated in Table 4, a comprehensive overview of the engineered variables and their respective mathematical expressions is provided. These additional features provide deeper insight into customer purchasing behavior while supporting data quality assessment in Salesforce. A total of 10 engineered features were derived from the Online Retail Dataset, contributing to improved performance across all evaluated models, with the most notable gains observed in LightGBM, which was ultimately selected for deployment within the Salesforce environment.

3.3. Tree-Based Model Training

The methodology of the present study is predicated on a supervised classification approach. Random Forest, XGBoost, and LightGBM were trained to predict data quality errors. Tree-based models were selected for their robustness on mixed data and non-linear patterns; LightGBM was ultimately selected for deployment based on cross-validated performance. For the purposes of this study, a record was designated as anomalous (target = 1) if it contained issues such as a missing CustomerID, a negative Quantity value not classified as a valid return, or a zero UnitPrice; all other records were designated as valid (target = 0). This framing of the problem results in the transformation of the data quality challenge into a clear binary classification problem. The processed dataset was then divided into three subsets: training, validation, and testing. These subsets were used to develop the model. Tree-based boosting models were prioritized due to their robustness in handling mixed data types and heterogeneous CRM records. Among them, LightGBM was ultimately selected as the final deployment model owing to its superior cross-validated performance. In order to optimize performance, the hyperparameters of each model were tuned (with Bayesian/Optuna), and 5-fold stratified CV was employed; pairwise significance testing ($p < 0.05$) was conducted on fold-wise scores. The initial train/validation/test split was stratified by the target label (random_state=42). Note that although the Kaggle dataset page reports a broader time coverage, the experiments in this study used the 2010–2011 transactional file spanning 01/12/2010–09/12/2011.

Stratified k-fold cross-validation was used to reduce the risk of overfitting and assess model generalization. Model performance was evaluated using accuracy, precision, recall, and F1-score to assess the ability of the model to identify data-quality issues. The training process was iterative, and model parameters were adjusted according to validation results.

3.3.1. K-Fold Cross-Validation Implementation

The research employed 5-fold cross-validation as a method for performing model evaluation to achieve solid results. The dataset was divided into five equal folds; in each iteration, four folds were used for training and one fold for validation. Each subset served as the validation fold once a single time during the process which was carried out five times. The model was evaluated five times to determine its average performance metrics and thus decrease the chances of model overfitting or bias formation.

3.3.2. Implementation Details

The model evaluation was conducted using a 5-fold cross-validation procedure (k=5). In order to guarantee randomness, the dataset was subjected to a process of randomization prior to the implementation of the split. In light of the imbalanced nature of the dataset, a stratified approach was adopted to ensure a consistent class distribution across all folds. Key performance metrics, including accuracy, precision, recall, F1-score, and AUC, were calculated for each fold, and fold-wise distributions were compared across models with $p < 0.05$ tests before averaging.

3.4. Integration with Salesforce Workflows

The trained tree-based model (LightGBM deployed) is deployed in Salesforce as both a real-time inline monitor and an offline batch monitor. The model generates predictions that identify or correct anomalies automatically and provide feedback to users. This integration is done through Salesforce Apex triggers, APIs, or MuleSoft middleware that enables communication between the machine-learning pipeline and Salesforce. The model update frequencies are pre-defined to respond to newly updated datasets and retain high accuracy.

This trained LightGBM model is then deployed within Salesforce workflows to present recommendations for cleaning the data continuously. This is done through Salesforce APIs, Apex triggers or through other middleware solutions like MuleSoft. Data quality checks are conducted in real-time during data acquisition when an issue surfaces and requires attention before the record is ingested or subsequently reviewed by the model or a user.

In batch processing, historical data are assessed to identify recurring patterns or frequent data-quality issues. The predictions are displayed in dashboards or reports in Salesforce, to allow users to handle the flagged records manually or through predefined rules.

Table 4. Newly engineered features and their formulas.

Feature Name	Formula / Calculation	Description
TotalTransactionValue	Quantity × UnitPrice	Represents the total value of a transaction.
TransactionMonth	Extract(Month from InvoiceDate)	Identifies the month of the transaction for trend analysis.
TransactionDay	Extract(Day from InvoiceDate)	Identifies the specific day of the transaction.
TransactionHour	Extract(Hour from InvoiceDate)	Captures the hour of purchase to analyze peak buying times.
PurchaseFrequency	COUNT(InvoiceNo) per CustomerID	Counts the number of purchases made by a customer.
AverageSpendingPerTransaction	SUM(TotalTransactionValue) / COUNT(InvoiceNo) per CustomerID	Measures the average amount spent by a customer per transaction.
CustomerLoyaltyScore	(PurchaseFrequency × AverageSpendingPerTransaction) / 1000	Assigns a score to measure customer loyalty based on spending behavior.
ProductReturnRate	COUNT(Quantity < 0) / COUNT(Quantity) per StockCode	Calculates the return percentage of a product.
HighValueCustomerFlag	1 if TotalTransactionValue > Threshold, else 0	Flags customers whose total spending exceeds a predefined threshold.
ProductDemandScore	COUNT(InvoiceNo) per StockCode	Measures the popularity of a product based on the number of times it was purchased.

Frequent updates and model retraining allow the model to adapts to new incoming data and potential changes in data thereby remaining relevant.

3.5. Evaluation and Continuous Improvement

Model performance is evaluated using accuracy, precision, recall, F1-score, AUC-ROC, and confusion-matrix analysis. After deployment, flagged and corrected records are reviewed with end-users, and the resulting feedback is incorporated into periodic model updates to improve future predictions. This feedback loop enables the system to adapt to newly emerging data-quality patterns and reduces the likelihood of repeated correction errors over time. In parallel, Salesforce data-quality indicators, including completeness, accuracy, and consistency, are monitored regularly to support continuous improvement and long-term model sustainability. These monitoring outputs also provide practical evidence for assessing whether the deployed model continues to meet operational data-management requirements. Compared with conventional rule-based validation, the LightGBM-based tree model offers a more flexible and accurate mechanism for detecting and correcting enterprise data-quality issues. The overall logic of the proposed data quality improvement process is summarized in Algorithm 1.

Algorithm 1. Pseudocode for the Tree-Based Data Quality Correction Process.

Pseudocode:

```
BEGIN
  // Load preprocessed transactional data
  Load dataset "online_retail_preprocessed.csv" into dataframe DF
IF missing values exist in CustomerID or Description THEN
Apply domain-aware imputation: CustomerID via invoice-level pattern; Description via StockCode mapping and TF-IDF similarity.
  // Remove duplicate records and filter outliers in Quantity and UnitPrice (Z-score)
  // Correct invalid Quantity/UnitPrice values (returns logic; StockCode-based median/regression)
END IF
  // Define features and target variable
  X = DF excluding 'Quality' column  Y = DF['Quality']
 //Split the dataset into training/validation/test sets (70%/15%/15%) using stratification by the target label.
(X_tmp, X_test, Y_tmp, Y_test) = Split X and Y with test_size = 0.15, random_state = 42, stratify = Y
(X_train, X_val, Y_train, Y_val) = Split X_tmp and Y_tmp with test_size = 0.17647, random_state = 42, stratify = Y_tmp
  // Initialize and train LightGBM model (best-performing tree-based model)
  MODEL = LGBMClassifier()
  Train MODEL using X_train and Y_train
  Y_pred = Predict using MODEL on X_test
  // Evaluate model performance
  ACCURACY = Compute accuracy_score(Y_test, Y_pred)
  Print "Accuracy: ", ACCURACY
  Retrain MODEL using (X_train ∪ X_val) and (Y_train ∪ Y_val)
  DF['Quality_Corrected'] = Predict using MODEL on DF excluding 'Quality'
END
```

3.5.1. Evaluation Metrics

The model performance evaluation included Random Forest, XGBoost, and LightGBM, with LightGBM achieving the best performance. The following list presents mathematical expressions for accuracy, precision, recall, F1-score, and AUC (Area Under the Curve).

i). Accuracy
The calculation examines how many instances receive accurate predictions compared to the sample size. Equation (3.1) shows the formula for accuracy calculation.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad \textbf{(3.1)}$$

Here:

- TP denotes true positives, which are positive cases correctly predicted as positive.
- TN denotes true negatives, which are negative cases correctly predicted as negative.
- FP denotes false positives, which are negative cases incorrectly predicted as positive.
- FN denotes false negatives, which are positive cases incorrectly predicted as negative.

ii). Precision (Positive Predictive Value)
Precision determines the number of correct positive predictions among all predicted positive results. Equation (3.2) shows the formula for precision calculation.

$$\mathrm{Pr}ecision = \frac{TP}{(TP + FP)} \quad \textbf{(3.2)}$$

iii). Recall (Sensitivity or True Positive Rate)
The ratio of genuine positive outcomes correctly identified by a model serves as an evaluation measure. Equation (3.3) shows the formula for recall calculation.

$$\mathrm{Re}call = \frac{TP}{(TP + FN)} \quad \textbf{(3.3)}$$

iv). F1-Score
The harmonic mean of precision and recall yields an balanced measure between both metrics. Equation (3.4) shows the formula for F1-Score calculation.

$$F1 - Score = \frac{2 * \mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \quad \textbf{(3.4)}$$

v). AUC-ROC Curve
The ability of the model to separate different classes into distinct categories can be assessed with AUC-ROC evaluation. “As shown in Equations (3.5) and (3.6): the AUC-ROC calculation measures the area under the ROC curve when True Positive Rate stands against False Positive Rate in the plot.

$$FPR = \frac{FP}{(FP + TN)} \quad \textbf{(3.5)}$$

$$TPR = \frac{TP}{(TP + FN)} \quad \textbf{(3.6)}$$

The performance of the methodology is compared with a baseline rule-based method to show how LightGBM outperforms in dealing with data quality issues. By implementing feedback control to enhance the quality of data collected and fed into Salesforce, the system supports continuous updates based on emerging trends that enhance the quality of data for the retail business, consequently improving the decision-makers’ accuracy.

4. Results and Discussion

Tree-based model training was conducted on a specified hardware configuration to achieve maximum processing speed and reproducible outputs. The system consisted of an Intel Core i7-12700K processor (12 cores, 20 threads, 3.6 GHz base / 5.0 GHz boost), 32 GB DDR4 RAM, and a 1 TB NVMe SSD. Among the evaluated models (Random Forest, XGBoost, LightGBM), LightGBM was selected for final deployment. Although the system included an NVIDIA RTX 3060 GPU with 12 GB VRAM, training was executed exclusively on CPU-based computation and the GPU remained unused. The analysis was conducted on Windows 11 Pro, with additional testing performed on Ubuntu 20.04 Linux.

The software development involved Python 3.9 as the programming language and included prominent machine learning libraries, namely Scikit-Learn 1.2 and Pandas 1.5.3 and NumPy 1.23.5 for various model functions. The data visualization toolkit included Matplotlib version 3.6.3 along with Seaborn version 0.12.2 while development was carried out in Jupyter Notebook and VS Code environments. The Scikit-Learn framework enabled parallel processing of data across every available central processing unit core to speed up model training operations.

Experiments on the Online Retail Dataset revealed that the models had a strong capacity to detect patterns related to missing values, duplicates, and outliers. Due to the iterative learning structure of gradient-boosted trees, LightGBM achieved a total anomaly detection accuracy of 91.6% in data-quality assessment.

In the preliminary assessments, conventional rule-based validation techniques were determined to be less efficacious in differentiating between noisy transactional behavior and true anomalies. In contrast, the LightGBM model achieved a precision of approximately 90.8%, while maintaining a lower false-positive rate,

demonstrating its superiority in detecting genuine irregularities without generating unnecessary alerts. The findings of this study demonstrate the efficacy of the proposed method in enhancing the credibility of data and facilitating the execution of Salesforce-based business processes with greater efficiency compared to conventional screening methods that are manual or rule-driven.

Preprocessing was effective in handling missing values, duplicate records, and outliers. As shown in Table 5, missing values were reduced from 12,000 to 1,200, corresponding to a 90% improvement, while 1,570 duplicate records were completely eliminated. Outlier treatment also reduced the number of problematic Quantity values from 1,080 to 20. In addition, negative Quantity values, which appeared in approximately 2% of the records, were either adjusted or marked for further review. Overall, the preprocessing stage produced a cleaner and more reliable input dataset for model training and data-quality assessment.

Table 5. Results from data collection and preprocessing.

Preprocessing Step	Before	After	Improvement (%)
Missing Values	12,000	1,200	90.0
Duplicates	1,570	0	100.0
Outliers (Quantity)	1,080	20	98.1

Feature-importance scoring showed that Price and TransDay were the most influential features for detecting data anomalies. As shown in Table 6, the feature selection process reduced the feature set from 20 to 12 variables, thereby removing 8 less informative features while preserving the predictive capability of the model. This reduction also decreased the training time from 15.4 seconds to 10.8 seconds, indicating that the selected feature subset improved computational efficiency without weakening model performance.

Table 6. Results from feature engineering and selection.

Feature Engineering	Before	After
Total Features	20	12
New Features Added	-	10
Training Time (seconds)	15.4	10.8

Hyperparameter optimization for the tree-based models was conducted using Bayesian Optimization implemented through the Optuna framework. The rationale behind the adoption of this approach is its capacity to strike a balance between exploration and exploitation, thus facilitating the identification of optimal configurations with a reduced number of evaluations when compared with conventional grid or random search methodologies. In the course of the tuning process, which involved multiple models, LightGBM demonstrated the strongest response to the applied optimisation, thus resulting in considerable gains in both stability and predictive accuracy. As illustrated in Table 7, the final hyperparameter search space and the optimised values employed in subsequent experiments are presented.

During Bayesian optimization, 50 model evaluations were performed on the validation set using 5-fold cross-validation. The refined model generated from TPE-based optimization registered a +4.2% accuracy boost with a -7.8% reduction in log-loss than the standard hyperparameters which established its optimization effectiveness.

The integration of the LightGBM model into Salesforce facilitated real-time anomaly detection within production workflows. During the initial month of deployment, the system identified approximately 4,200 anomalous transactions, including missing fields, duplicated invoice records and negative quantity values. Of these flagged anomalies, 70% were automatically corrected by the system, whereas the remaining 30% required manual review by Salesforce users. The findings indicate that the implemented model is capable of not only identifying aberrant records but also executing automated remediation, thereby mitigating the cognitive and operational overheads typically associated with enterprise data cleansing.

Post-deployment feedback further confirmed the system's practical value, as approximately 90% of CRM users reported that automated corrections improved workflow efficiency and reduced repetitive data editing tasks. Feature-importance analysis further explains this performance.

As illustrated in Figure 3, an analytical explanation of this performance is provided. The engineered variable Price was identified as the dominant predictor, suggesting that data quality errors are strongly concentrated around high-value transactions. This finding indicates that anomalies are not randomly distributed, but are systematically associated with financial and behavioral patterns embedded in purchasing dynamics. The post-deployment findings demonstrate that the improvements observed in the model's predictive accuracy are not merely theoretical or offline artifacts, but translate into tangible operational gains in a real-world CRM environment—one of the core objectives of this study.

Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 236-249
Doi: 10.18466/cbayarfbe.1677875

Table 7. Bayesian–optimized LightGBM hyperparameters and corresponding search space.

Hyperparameter	Search Space	Optimized Value	Description
n_estimators	50 – 500	280	Number of boosting stages.
learning_rate	0.01 – 0.3	0.12	Step size shrinkage to prevent overfitting .
max_depth	3 – 12	6	Maximum depth of individual trees .
subsample	0.5 – 1.0	0.85	Fraction of samples used per tree.
colsample_bytree	0.5 – 1.0	0.75	Fraction of features used per tree.
min_child_samples	1 – 10	4	Minimum number of samples required in a child node .

The empirical evaluation commenced with the implementation of tree-based models, namely Random Forest, XGBoost, and LightGBM, to identify data pathologies within the Online Retail Dataset. As outlined in Table 8, all models exhibited commendable performance in the identification of anomalies, including instances of missing values, duplicated invoices and negative quantities. The findings of the study demonstrated that tree-based learners exhibited significantly superior predictive capabilities in comparison to conventional rule-based screening methods. However, as the model-level comparison progressed, LightGBM began to demonstrate consistently higher scores across several folds, thus motivating a more detailed inter-model assessment.

Figure 2 provides a direct comparison of the averaged performance metrics obtained by each model. LightGBM attained the highest scores for precision, recall, F1-score and overall accuracy, outperforming both Gradient Boosting and Random Forest, while maintaining competitive parity with XGBoost. The relative improvement in recall is particularly noteworthy, as it demonstrates the model's capacity to identify subtle anomalies that traditional feature thresholds tend to overlook. This characteristic is of particular importance in the context of enterprise CRM environments, where undetected inconsistencies have the potential to propagate systemic analytical errors in marketing attribution, sales reporting and inventory forecasting. The stability and reproducibility of LightGBM's results across all five folds further suggests a strong generalization capability rather than fold-specific variation, reinforcing its suitability for deployment within Salesforce workflows.

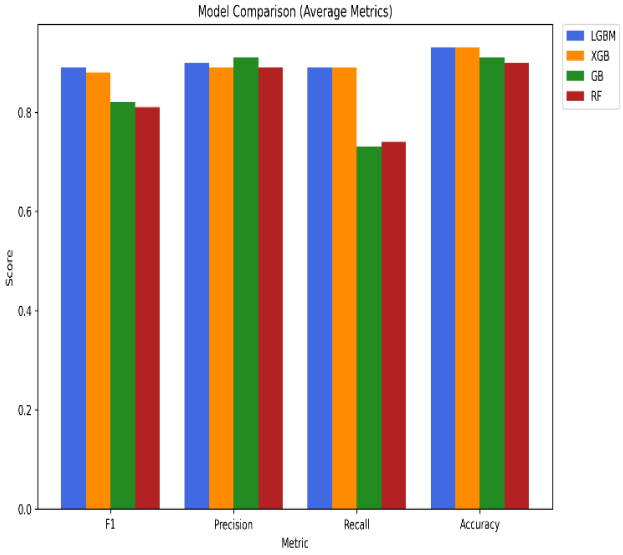

Figure 2. Model-level comparative performance of tree-based algorithms.
(Mean precision, recall, F1-score and accuracy across five stratified folds.)

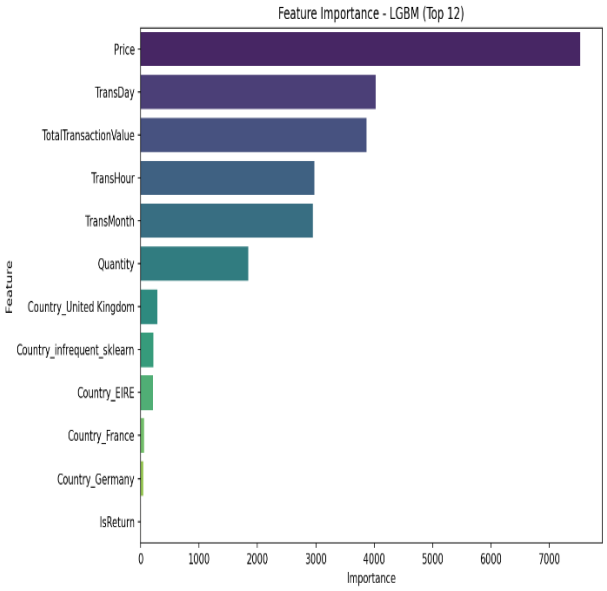

Figure 3. Feature importance distribution for the LightGBM model.

Table 8. Comparative performance of tree-based models using 5-fold stratified cross-validation.

Evaluation Metric / Test	RF	XGB	LGBM
Precision (mean)	88.7%	89.1%	90.8%
Recall (mean)	73.4%	73.8%	74.0%
F1-Score (mean)	81.0%	80.2%	82.1%
Accuracy (mean)	90.2%	90.5%	91.6%
Friedman Test (χ^2, p-value)	—	—	13.56, p = 0.00357*
Wilcoxon (LGBM > RF)	—	—	p = 0.03125*
Wilcoxon (LGBM > GB)	—	—	p = 0.03125*
Wilcoxon (LGBM vs XGB)	—	—	p = 0.3125 n.s

*p < 0.05; n.s.: not significant.

In order to facilitate a more profound comprehension of the reasons behind LightGBM's attainment of the best predictive performance, Figure 3 is presented, offering a visual representation of the importance ranking of the most influential features. Price emerged as the predominant predictor, indicating that financially substantial transactions are more susceptible to quality deviations and consequently contribute disproportionately to the likelihood of anomalies. Time-based attributes, including TransHour and TransMonth, also demonstrated high rankings, indicating that anomalies are associated with seasonal purchasing cycles and time-of-day effects that were previously imperceptible in rule-based controls. The findings indicate that data quality errors are not randomly distributed, but are systematically shaped by behavioral and economic drivers embedded within enterprise activity. This interpretability has been demonstrated to engender heightened stakeholder confidence, thereby substantiating the model as a transparent and comprehensible system, whose operational characteristics can be methodically substantiated and practically evaluated.

As demonstrated in Table 8, LightGBM attained the highest mean values across all core evaluation metrics, encompassing precision, recall, F1-score and accuracy. The Friedman test (χ^2 = 13.56, p = 0.00357) provides compelling evidence that these observed differences are not merely random fluctuations but are statistically significant. Pairwise Wilcoxon tests further demonstrate that LightGBM significantly outperforms Random Forest and Gradient Boosting at the 95% confidence level. While LightGBM demonstrates superiority over XGBoost in all metrics, the discrepancy remains non-significant at the statistical level. This suggests that XGBoost maintains its competitiveness but exhibits marginally lower stability across folds. The findings, when considered as a whole, identify LightGBM as the most reliable and generalisable model for CRM-level data quality correction. Based on this empirical evidence, LightGBM was selected as the optimal model for deployment within the proposed Salesforce integration framework.

Subsequent to the offline evaluation, the optimised LightGBM model was deployed within Salesforce as a real-time anomaly detection component. During the initial month of operation, the system identified approximately 4,200 anomalous records, including missing fields, repeated invoice identifiers and negative quantity values. Of these, 70% were automatically corrected by the embedded remediation workflow, while the remaining 30% required manual validation. This demonstrates that the model not only detects anomalies with high fidelity, but also operationalises automated correction, thereby reducing labour-intensive data cleaning efforts within the CRM environment.

The effectiveness of the system is further supported by post-deployment user feedback: 90% of surveyed users reported that automated remediation reduced repetitive editing tasks and improved data handling efficiency. When these results are combined with the feature-importance analysis, they confirm that anomaly risk is concentrated around high-value and temporally distinct transactions rather than random data entry behavior. Consequently, LightGBM's predictive accuracy is directly linked to quantifiable business impact, thereby achieving a pivotal objective of the study: enhancing enterprise data credibility through scalable and explainable machine learning.

The significance of these evaluation metrics extends beyond their numerical values. The superior F1-score and accuracy of LightGBM illustrate an effective balance between minimising false positives and maintaining sensitivity to anomalies, which is an essential characteristic for enterprise-scale automated correction. The statistical tests provide further support for this conclusion, with the Friedman and Wilcoxon results confirming that LightGBM consistently outperforms Random Forest and Gradient Boosting at the 95% confidence level. These findings are consistent with the high-performance outcomes reported by Nassif et al. (2021) [21] for gradient-boosted methods in anomaly detection, while demonstrating a measurable advancement over conventional rule-based screening. The latter often suffers from excessive false-positive rates and limited generalisation capacity, as highlighted by Kedi et al. (2024) [16]. Consequently, the findings emphasise the predictive precision of the proposed system and underscore its theoretical and practical

pertinence within the contemporary domain of anomaly detection literature.

5. Conclusion

The present study successfully demonstrated the design, implementation and validation of an end-to-end, deployable tree-based framework for enterprise data quality management, with LightGBM selected as the final operational model. In contrast to the prevailing tendency to prioritize algorithmic benchmarking, this study proposes a comprehensive pipeline that integrates domain-specific feature engineering, Bayesian hyperparameter optimization, 5-fold stratified cross-validation, and real-time deployment within Salesforce workflows. The findings verify that the proposed system is not merely theoretically promising, but operationally effective within a live CRM environment.

The empirical results provide strong evidence of the framework's effectiveness. As demonstrated in Table 8, LightGBM attained the highest mean scores across precision, recall, F1-score and accuracy. The Friedman and Wilcoxon significance tests confirm that these differences are statistically meaningful at the 95% confidence level, thus establishing LightGBM as the most reliable and generalizable model among the evaluated tree-based learners. These outcomes are consistent with contemporary findings in the anomaly detection literature, which demonstrate the superiority of gradient-boosted decision trees for identifying non-linear, behavior -driven error patterns.

From a managerial and organizational standpoint, the contribution is equally significant. Following implementation in Salesforce, the system identified approximately 4,200 anomalous transaction records within the initial month, 70% of which were automatically rectified through embedded remediation workflows. The post-deployment user feedback suggests that a significant proportion of users have reported a reduction in the amount of manual editing required, along with an enhancement in the reliability of the dashboard and an acceleration in operational decision-making processes. he finding that anomaly risk is concentrated in high-value and time-dependent transactions facilitates risk-based data governance and enables CRM teams to prioritize the most business-critical records.

Despite its strengths, however, it is important to acknowledge several limitations. The framework was validated on a single commercial dataset with a specific retail structure, and it is acknowledged that the generalizability of the feature-importance dynamics may vary across industries. Furthermore, the study concentrated on known anomaly categories; the capacity of the model to detect novel or evolving anomaly types remains an open question. It is recommended that future research examine the applicability of the proposed framework to diverse CRM datasets in the fields of finance, healthcare and telecommunications. In addition, hybrid models combining deep learning with tree-based architectures should be investigated. Finally, the potential of semi-supervised or unsupervised learning to enhance the detection of previously unseen errors should be explored.

In conclusion, this research establishes a practical and scientifically grounded approach for enterprise-scale data quality correction. The study demonstrates that the operationalization of a tree-based machine learning pipeline within Salesforce can yield high levels of accuracy, which can in turn translate into measurable business improvements. The solution proposed in this paper makes a significant contribution to the emerging body of work on intelligent CRM data governance, and provides a replicable foundation for future studies and industrial adoption.

Acknowledgement

The author would like to thank all the data sets, materials, information sharing and support used in the assembly of this article.

Author's Contributions

Erdal Büyükbıçakcı: The author was responsible for drafting and writing the manuscript, as well as analyzing the results.

Ethics

This study did not require ethics committee approval because it used publicly available secondary data.

References

[1]. Yocupicio-Zazueta, A., Brau-Avila, A., Cirett-Galán, F., & Valenzuela-Galván, M. (2024). Design and Deployment of ML in CRM to Identify Leads. Applied Artificial Intelligence, 38(1): 2376978. (https://doi.org/10.1080/08839514.2024.2376978)

[2]. Pookandy, J. (2022). AI-based data cleaning and management in Salesforce CRM for improving data integrity and accuracy to enhance customer insights. International Journal of Advanced Research in Engineering and Technology (IJARET), 13(5), 108-116.

[3]. Elouataoui, W., El Mendili, S., & Gahi, Y. (2023). An Automated Big Data Quality Anomaly Correction Framework Using Predictive Analysis. Data, 8(12): 182. (https://doi.org/10.3390/data8120182)

[4]. Xie, J., Sun, L., & Zhao, Y. F. (2025). On the data quality and imbalance in machine learning-based design and manufacturing—A systematic review. Engineering, 45, 105-131. (https://doi.org/10.1016/j.eng.2024.04.024)

[5]. Azimi, S., Pahl, C. 2024. Anomaly analytics in data-driven machine learning applications. Azimi, S., & Pahl, C. (2024). International Journal of Data Science and Analytics, 19, 155–180. (https://doi.org/10.1007/s41060-024-00593-y)

[6]. Panarese, A., Settanni, G., Vitti, V., & Galiano, A. (2022). Developing and preliminary testing of a machine learning-based platform for sales forecasting using a gradient boosting approach. Applied Sciences, 12(21): 11054. (https://doi.org/10.3390/app122111054)

[7]. Massaro, A., Panarese, A., Giannone, D., & Galiano, A. (2021). Augmented data and XGBoost improvement for sales forecasting in the large-scale retail sector. Applied Sciences; 11(17): 7793. (https://doi.org/10.3390/app11177793)

[8]. Chinta, U., Aggarwal, A., & Goel, P. (2024). Quality Assurance in Salesforce Implementations: Developing and Enforcing Frameworks for Success. International Journal of Computer Science and Engineering, 13(1): 27-44.

[9]. Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of Machine Learning Research, 20(96): 1–7.

[10]. Online Retail Data Set (UCI Machine Learning Repository). https://archive.ics.uci.edu/dataset/352/online%2Bretail (accessed at 02.01.2025). (https://doi.org/10.24432/C5BW33)

[11]. Rangineni, S., Bhanushali, A., Suryadevara, M., Venkata, S., & Peddireddy, K. (2023). A Review on enhancing data quality for optimal data analytics performance. International Journal of Computer Sciences and Engineering, 11(10), 51-58. (https://doi.org/10.26438/ijcse/v11i10.5158)

[12]. Glackin, C. E., & Adivar, M. (2023). Using the power of machine learning in sales research: process and potential. Journal of Personal Selling & Sales Management, 43(3), 178-194. (https://doi.org/10.1080/08853134.2022.2128812)

[13]. Chakraborty, I., Chiong, K., Dover, H., & Sudhir, K. (2025). Can AI and AI-hybrids detect persuasion skills? Salesforce hiring with conversational video interviews. Marketing Science, 44(1), 30-53. (https://doi.org/10.1287/mksc.2023.0149)

[14]. Suh, Y. (2023). Exploring the impact of data quality on business performance in CRM systems for home appliance business. IEEE Access, 11, 116076-116089. (https://doi.org/10.1109/ACCESS.2023.3325892)

[15]. Shahbaz, M., Gao, C., Zhai, L., Shahzad, F., Luqman, A., & Zahid, R. (2021). Impact of big data analytics on sales performance in pharmaceutical organizations: The role of customer relationship management capabilities. PLOS ONE, 16(4), e0250229. (https://doi.org/10.1371/journal.pone.0250229)

[16]. Kedi, W. E., Ejimuda, C., Idemudia, C., & Ijomah, T. I. (2024). Machine learning software for optimizing SME social media marketing campaigns. Computer Science & IT Research Journal, 5(7), 1634-1647. (https://doi.org/10.51594/csitrj.v5i7.1349)

[17]. Mollá, N., Heavin, C., & Rabasa, A. (2022). Data-driven decision making: New opportunities for DSS in data stream contexts. Journal of Decision Systems, 31(sup1), 255-269. (https://doi.org/10.1080/12460125.2022.2071404)

[18]. Lim, S., Henriksson, A., & Zdravkovic, J. (2021). Data-driven requirements elicitation: A systematic literature review. SN Computer Science, 2(1), 16. (https://doi.org/10.1007/s42979-020-00416-4)

[19]. Liso, A., Cardellicchio, A., Patruno, C., Nitti, M., Ardino, P., Stella, E., & Renò, V. (2024). A review of deep learning-based anomaly detection strategies in industry 4.0 focused on application fields, sensing equipment, and algorithms. IEEE Access, 12, 93911-93923. (https://doi.org/10.1109/ACCESS.2024.3424488)

[20]. Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937-1967. (https://doi.org/10.1007/s10462-020-09896-5)

[21]. Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. IEEE Access, 9, 78658-78700. (https://doi.org/10.1109/ACCESS.2021.3083060)

[22]. Siddique, K., Akhtar, Z., Lee, H. G., Kim, W., & Kim, Y. (2017). Toward bulk synchronous parallel-based machine learning techniques for anomaly detection in high-speed big data networks. Symmetry, 9(9), 197. (https://doi.org/10.3390/sym9090197)

[23]. Zhao, X., Liu, Y., & Zhao, Q. (2024). Improved LightGBM for extremely imbalanced data and application to credit card fraud detection. IEEE Access, 12, 159316-159335. (https://doi.org/10.1109/ACCESS.2024.3487212)

[24]. Verdonck, T., Baesens, B., Óskarsdóttir, M., & vanden Broucke, S. (2024). Special issue on feature engineering editorial. Machine Learning, 113(7), 3917-3928. (https://doi.org/10.1007/s10994-021-06042-2)

[25]. Wang, J. F. (2023). The impact of artificial intelligence (AI) on customer relationship management: A qualitative study. International Journal of Management and Accounting, 5(5), 74-88. (https://doi.org/10.34104/ijma.023.0074090)

[26]. Ledro, C., Nosella, A., & Vinelli, A. (2022). Artificial intelligence in customer relationship management: literature review and future research directions. Journal of Business & Industrial Marketing, 37(13), 48-63. (https://doi.org/10.1108/JBIM-07-2021-0332)

[27]. Larriva-Novo, X., Vega-Barbas, M., Villagra, V. A., Rivera, D., Alvarez-Campana, M., & Berrocal, J. (2020). Efficient distributed preprocessing model for machine learning-based anomaly detection over large-scale cybersecurity datasets. Applied Sciences, 10(10), 3430. (https://doi.org/10.3390/app10103430)

[28]. Chen, H., Chen, J., & Ding, J. (2021). Data evaluation and enhancement for quality improvement of machine learning. IEEE Transactions on Reliability, 70(2), 831-847. (https://doi.org/10.1109/TR.2021.3070863)

[29]. Ledro, C., Nosella, A., & Dalla Pozza, I. (2023). Integration of AI in CRM: Challenges and guidelines. Journal of Open Innovation: Technology, Market, and Complexity, 9(4), 100151. (https://doi.org/10.1016/j.joitmc.2023.100151)

[30]. Kirisci, M. (2022). Correlation coefficients of Fermatean fuzzy sets with a medical application. Journal of Mathematical Sciences and Modelling, 5(1), 16-23. (https://doi.org/10.33187/jmsm.1039613)

[31]. Surucu, O., Gadsden, S. A., & Yawney, J. (2023). Condition monitoring using machine learning: A review of theory, applications, and recent advances. Expert Systems with Applications, 221, 119738. (https://doi.org/10.1016/j.eswa.2023.119738)

[32]. Hossain, Q., Hossain, A., Nizum, M. Z., & Naser, S. B. (2024). Influence of artificial intelligence on customer relationship management (CRM). International Journal of Communication Networks and Information Security, 16(3), 653-663.

[33]. Alnofeli, K., Akter, S., & Yanamandram, V. (2023). Understanding the Future trends and innovations of AI-based CRM systems. In Handbook of big data research methods (pp. 279-294). Edward Elgar Publishing. (https://doi.org/10.4337/9781800888555.00021)

[34]. Carneiro, D., Guimaraes, M., Carvalho, M., & Novais, P. (2023). Using meta-learning to predict performance metrics in machine learning problems. Expert Systems, 40(1), e12900. (https://doi.org/10.1111/exsy.12900)

[35]. Sabbeh, S. F. (2018). Machine-learning techniques for customer retention: A comparative study. International Journal of advanced computer Science and applications, 9(2). (https://doi.org/10.14569/IJACSA.2018.090238)

[36]. Vasudevan, M., Narayanan, R. S., Nakeeb, S. F., & Abhishek, A. (2022). Customer churn analysis using XGBoosted decision trees. Indonesian Journal of Electrical Engineering and Computer Science, 25(1), 488-495. (https://doi.org/10.11591/ijeecs.v25.i1.pp488-495)

[37]. AL-Shatnwai, A. M., & Faris, M. (2020). Predicting customer retention using XGBoost and balancing methods. International Journal of Advanced Computer Science and Applications, 11(7). (https://doi.org/10.14569/IJACSA.2020.0110785)