A Scalable Machine Learning Framework for Intelligent Personal Financial Decision Support

SAHAR HOOSHMAND[1], MALLIKARJUNA YERUKU[1], BENYAMIN AHMADNIA[1], ALI JALOOLI[1]

[1]Department of Computer Science, California State University-Dominguez Hills, Carson, CA, USA

E-MAIL: hooshmand@csudh.edu, myeruku1@toromail.csudh.edu, bahmadniayebosari@csudh.edu, ajalooli@csudh.edu

Abstract:

Finance plays a central role in everyday life, shaping decisions around spending, saving, and borrowing across activities such as banking transactions, credit card use, and loan management. This paper presents a scalable machine learning framework for intelligent personal financial applications, leveraging established models including Linear Regression, Gradient Boosting, and TF-IDF with k-Nearest Neighbors. The proposed system analyzes historical transaction data to track income and expenditure patterns, detect recurring payments, generate personalized monthly budgets, and provide cost-effective purchase recommendations. Experimental evaluation demonstrates strong predictive performance, with an income forecasting error of approximately 2.5%, an expense forecasting error of about 23% with calibrated uncertainty bounds, category budget prediction explaining 68.5% of the variance (R^2), and coupon matching achieving $\geq$95% similarity for valid products. The results indicate that lightweight, interpretable machine learning models can deliver reliable financial forecasting and budgeting insights, motivating future work on richer datasets and advanced temporal modeling for improved personalization.

Keywords:

Personal finance management, machine learning, time-series forecasting, expense prediction, budget analytics, recommendation systems, financial data analysis.

1. Introduction

Personal financial management is essential for economic stability, long-term planning, and individual well-being. With the rapid growth of digital banking, users generate large volumes of transaction data across savings accounts, credit cards, loans, and digital payment platforms. Despite this, most personal finance applications remain largely descriptive, offering balance summaries and historical transaction views with limited predictive intelligence. As a result, users lack effective tools to analyze spending behavior, identify recurring expenses, and make informed budgeting decisions, particularly when financial data is distributed across multiple institutions. At the initial stage, the system focuses on building a comprehensive transaction dataset by collecting financial records from multiple heterogeneous sources, including user-entered transactions, synthetically generated financial data, and sandbox-based banking data retrieved through the Plaid API. The collected dataset captures essential transaction attributes such as transaction amount, timestamp, merchant name, category identifier, and income or expense indicators, forming a unified and structured foundation for downstream analytics and machine learning–based forecasting.

This paper presents an AI-driven personal finance management system [1] that addresses these challenges by integrating machine learning–based forecasting, category-level budgeting, and recommendation intelligence within a secure and scalable web architecture [2]. The system aggregates multi-account financial data using Plaid API integration and processes both user-entered and synthetic transaction records. Time-series regression models [3, 4] are applied for income and expense forecasting, ensemble learning techniques are used for category-level budget prediction, and a hybrid TF-IDF [5, 6] and k-Nearest Neighbors [7, 6] framework enables personalized coupon recommendations. By combining predictive analytics with interactive dashboards and privacy-aware system design [8], the proposed platform transforms personal finance management from passive monitoring into an intelligent, data-driven decision-support system.

2. Related Work

Early academic and commercial systems primarily focused on account aggregation and transaction visualization, enabling users to monitor balances and historical spending patterns. More recent research has introduced machine learning techniques to enhance financial forecasting and behavioral analysis [9].

Ni et al. demonstrated that regression-based models are effective for short-term income and expense prediction when financial patterns are stable and noise levels are low [3]. However, linear models often struggle with non-linear spending behavior, seasonal variation, and category-level irregularities. To address these challenges, ensemble learning methods such as Gradient Boosting [10] [11] have been applied to capture complex temporal relationships and feature interactions in financial data. Time-series regression [3] [4] with lagged features and rolling statistics offers a practical balance between interpretability and predictive accuracy, while deep learning approaches such as LSTMs [12] provide greater modeling capacity at the expense of higher data requirements and computational overhead.

Liu et al. investigated text-based similarity methods for recommendation tasks [5] and showed that TF-IDF combined with k-Nearest Neighbors can effectively support coupon and offer recommendation. Text-based similarity methods using TF-IDF [5] [6] and cosine similarity have been widely adopted for transaction categorization and offer matching, enabling personalized discount and coupon recommendations. Hybrid approaches that combine unsupervised similarity learning with domain-specific constraints have been shown to reduce false positives and improve recommendation precision in noisy datasets.

Our proposed model is distinguished by its integrated framework that combines time-series forecasting, ensemble-based category budgeting, and hybrid recommendation models within a secure, scalable, and user-centric personal finance management system designed for real-world deployment.

3. Methodology

3.1 System Design

The proposed personal finance management system [1] adopts a modular three-tier architecture to ensure scalability, security, and seamless integration with machine learning components. The backend is implemented using Python and Django with Django REST Framework, providing secure RESTful APIs and JWT-based authentication for user-level access control, while PostgreSQL is used for reliable and efficient financial data storage. The frontend is developed using React to deliver an interactive dashboard for financial visualization. External bank account integration is achieved through the Plaid API, enabling secure synchronization of sandbox-based transaction data. The system is organized into authentication, accounts, transactions, budgeting, and recommendation modules, supporting both manual and automated data ingestion. Transaction data is collected through a hybrid pipeline combining user-entered records, synthetic data generation, and Plaid-based synchronization, all stored under a unified schema to support consistent preprocessing and downstream machine learning workflows for forecasting and recommendation tasks.

3.2 Income Forecast Model

The Income Forecast Model is designed to estimate a user's next-month total income using transaction-level data collected through the web application and synchronized banking sources. Income transactions are identified using a binary *isIncome* flag assigned at ingestion time and stored in the PostgreSQL database. Let $\mathcal{I}_t = \{a_{t,1}, a_{t,2}, \dots, a_{t,n_t}\}$ denote the set of income transaction amounts recorded for a user during month t, where each $a_{t,i}$ originates from either manual entry or Plaid-synchronized data. These transaction-level records are aggregated into a univariate monthly income series:

$$y_t = \sum_{i=1}^{n_t} a_{t,i}, \tag{1}$$

which reduces intra-month noise and aligns the data with the monthly forecasting objective. Prior to aggregation, timestamps are normalized, duplicate records are removed, missing values are imputed, and extreme one-time inflows are capped using percentile-based thresholds to prevent distortion of regression coefficients. Exploratory analysis using rolling means and variance confirms income stability with limited volatility, justifying a regression-based temporal model.

Temporal dependencies are modeled using lagged features derived from the aggregated income series. For each month t, the feature vector is defined as:

$$\mathbf{x}_t = [1,\ y_{t-1},\ y_{t-2},\ y_{t-3}]^\top, \tag{2}$$

where the intercept captures baseline income and the lag terms encode short- and medium-term temporal dependence. The predicted income $\hat{y}_t$ is obtained using a Linear Regression [13] model:

$$\hat{y}_t = \mathbf{w}^\top \mathbf{x}_t, \tag{3}$$

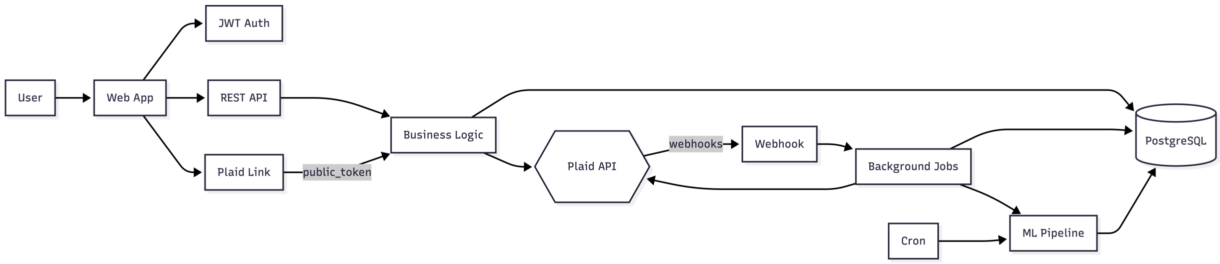

FIGURE 1. System Design Architecture

with parameters $\mathbf{w}$ estimated via ordinary least squares by minimizing:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{t \in \mathcal{T}} \left(y_t - \mathbf{w}^\top \mathbf{x}_t \right)^2, \tag{4}$$

where $\mathcal{T}$ denotes the set of training months. Model training and evaluation are performed using time-aware cross-validation with *TimeSeriesSplit* [3] [4], ensuring strict temporal ordering between training and validation sets and preventing data leakage. Forecast accuracy is assessed using Mean Absolute Error (MAE), providing an interpretable measure of monetary deviation and enabling reliable short-term income forecasting for budgeting and cash-flow planning.

3.3 Expense Forecast Model

The Expense Forecast Model predicts a user's next-month total expenditure using transaction-level data collected through the web application and synchronized financial accounts. Expense transactions are identified using categorical labels and an *isIncome* flag stored at ingestion time in the backend database. Let $\mathcal{E}_t = \{b_{t,1}, b_{t,2}, \ldots, b_{t,m_t}\}$ denote the set of expense transaction amounts recorded during month t, where each $b_{t,i}$ originates from either manual user entry or Plaid-synchronized data. Monthly total expense is computed as

$$z_t = \sum_{i=1}^{m_t} b_{t,i}, \tag{5}$$

transforming high-frequency transaction records into a univariate monthly expense series suitable for time-series modeling. Prior to aggregation, transaction timestamps are normalized, invalid or duplicate records are removed, income and transfer transactions are excluded, and extreme outliers are clipped to reduce the influence of irregular purchases. Exploratory analysis using rolling statistics confirms significant volatility and seasonality, motivating a regularized regression approach.

Temporal dependencies are modeled by constructing a feature vector $\mathbf{x}_t$ comprising lagged expenses from the previous 12 months, rolling mean statistics over 3-, 6-, and 12-month windows, and a calendar month indicator to encode seasonality. These features are standardized and used as inputs to a Ridge Regression model [14], which estimates future expenses while mitigating multicollinearity among correlated lag features. Model parameters are learned by minimizing the regularized objective

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{t \in \mathcal{T}} \left(z_t - \mathbf{w}^\top \mathbf{x}_t \right)^2 + \alpha \|\mathbf{w}\|_2^2, \tag{6}$$

where α controls the bias–variance trade-off. Hyperparameter selection is performed using `RidgeCV` with $\alpha \in \{10^{-3}, \ldots, 10^3\}$ under a rolling *TimeSeriesSplit* cross-validation scheme [3, 4] to prevent temporal leakage. Forecast accuracy is evaluated using Mean Absolute Error (MAE), ensuring stable and interpretable expense prediction suitable.

3.4 Category Budget Prediction Model

The Category Budget Prediction Model forecasts next-month spending for individual expense categories using transaction-level, Expense transactions are labeled with category identifiers at ingestion time and stored in the backend database. Let $\mathcal{C}$ denote the set of spending categories and let $\mathcal{E}_{c,t} = \{e_{c,t,1}, \ldots, e_{c,t,k_{c,t}}\}$ represent the set of expense transactions for category c during month t, originating from either manual entry or Plaid-based synchronization. Monthly category spending is computed as

$$s_{c,t} = \sum_{i=1}^{k_{c,t}} e_{c,t,i}, \tag{7}$$

producing category-specific monthly time series suitable for forecasting. Prior to aggregation, timestamps are normalized, duplicate and invalid records are removed, income transactions are excluded, and extreme outliers are mitigated to stabilize learning. Exploratory analysis confirms heterogeneous temporal behavior across categories, motivating a non-linear modeling approach.

For each category c, temporal features are constructed from $\{s_{c,t}\}$, including lagged spending values, rolling mean and standard deviation statistics, and calendar month indicators to encode seasonality. Category-level forecasts are generated using a Gradient Boosting Regressor [10, 11], which models spending as an additive ensemble of regression trees:

$$\hat{s}_{c,t} = \sum_{m=1}^{M} \gamma_m h_m(\mathbf{x}_{c,t}), \tag{8}$$

where $h_m(\cdot)$ denotes the m-th weak learner and $\mathbf{x}_{c,t}$ is the engineered feature vector for category c at time t. Model hyperparameters include the number of estimators ($M = 400$), learning rate ($\eta = 0.05$), maximum tree depth ($d = 3$), and subsampling ratio (0.9), balancing bias and variance. Training and evaluation employ time-aware cross-validation [3, 4] to preserve chronological order, with performance assessed using Mean Absolute Error (MAE), enabling robust category-level budget forecasting across both stable and highly variable spending categories.

3.5 Coupon Recommendation Model

The Coupon Recommendation Model maps user purchase transactions collected through the web application to relevant digital coupons using a lightweight, unsupervised text-similarity framework. Purchase data are extracted from expense transactions, where merchant names and transaction descriptions represent purchased items, while coupon data consist of textual attributes such as title, display description, long description, category, and brand metadata. All text fields are normalized via lowercasing, punctuation removal, stop-word filtering, and lemmatization to reduce linguistic variability. Each purchase item p and coupon c is embedded into a TF-IDF vector space [5, 6], and semantic similarity is computed using cosine similarity [15]. Candidate coupons are retrieved using a k-Nearest Neighbors (kNN) search [7, 6], enabling efficient matching without requiring labeled training data.

Coupon relevance is computed using a hybrid scoring function that combines semantic similarity and brand consistency:

$$\text{Score}(p, c) = \lambda \cdot \cos(\mathbf{v}_p, \mathbf{v}_c) + (1 - \lambda) \cdot I_{\text{brand}}(p, c), \tag{9}$$

where $\mathbf{v}_p$ and $\mathbf{v}_c$ denote the TF-IDF vectors of the purchase item and coupon, respectively, and $I_{\text{brand}}(p, c)$ indicates normalized brand token overlap. The weighting factor is empirically set to $\lambda = 0.8$. The TF-IDF vectorizer is configured with a maximum vocabulary of 5,000 features, an n-gram range of 1–2, and sublinear term-frequency scaling, while the kNN stage uses cosine distance with $k = 5$. A minimum cosine similarity threshold of 0.80 and coupon expiration filtering are enforced to reduce false positives. Model performance is evaluated using standard retrieval metrics including Precision@K, Recall@K, and Top-K accuracy, ensuring precise and computationally efficient coupon recommendations suitable for real-time deployment.

3.6 Proposed Pipeline

The proposed system Figure 2 employs an end-to-end, model-agnostic machine learning pipeline that converts transaction-level data collected through the web application into predictive financial insights. Transactions originating from user input, synthetically generated records, and Plaid-synchronized banking data are ingested into a centralized database with owner-based access control and include attributes such as amount, timestamp, merchant, category, and income/expense indicators. Prior to modeling, timestamps are normalized, missing and duplicate records are removed, and extreme outliers are clipped using robust statistical thresholds. Exploratory analysis using monthly aggregation and rolling statistics informs feature selection and model configuration. Feature engineering extracts lagged values, rolling means, seasonal indicators for time-series forecasting, and TF-IDF–based text embeddings [5, 6] for recommendation tasks. Models-including linear regression [13], regularized and ensemble regressors, and similarity-based retrieval-are trained using time-aware cross-validation to prevent temporal leakage, with hyperparameters optimized via logarithmic or grid-based search. Performance is evaluated using task-specific metrics such as MAE for regression and Precision@K or Recall@K for recommendations, ensuring stable generalization and deployability.

4. Experimental Results

In the following section, we detail the experimental setup, performance evaluation, and insights we gained from applying this pipeline to financial transactions.

4.1 Dataset

The experimental dataset comprises approximately 1,600 transactions collected over 24 months from a single modeled

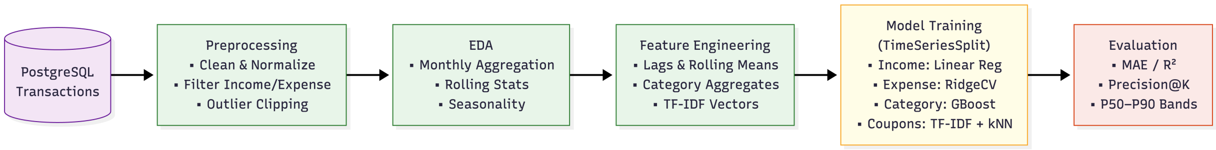

FIGURE 2. Proposed Machine Learning Pipeline

user. Data were obtained from seeded synthetic generation (50%), Plaid Sandbox API (35%), and user-entered transactions (15%). For forecasting tasks, transactions were aggregated monthly, yielding 24 time points per user. A chronological 80/20 split (19 training months, 5 testing months) was applied to prevent data leakage. Additionally, rolling-origin cross-validation was used to simulate real-world deployment. Model performance was evaluated against naïve last-month and 3-month moving average baselines using MAE, MAPE, and R^2 for regression tasks.

TABLE 1. Dataset Statistics

Dataset Attribute	Value
Total number of transactions	∼1,600
Time span	24 months
Average transactions per month	∼65
Number of expense categories	>20
Income transactions	∼18%
Expense transactions	∼82%
Recurring transactions	∼35%
Unique merchants	>150
Text-based features	Merchant name, description
Numerical features	Amount, lag values, rolling means
Temporal features	Month, year, rolling windows

4.2 Model Implementation

All machine learning models were implemented in Python using the `scikit-learn` framework, with data processing and feature engineering performed using `pandas` and `NumPy`. Time-aware validation strategies were consistently applied across forecasting tasks to ensure temporal correctness and prevent data leakage. [1]

The Income Forecast Model uses `LinearRegression` [13] with monthly income aggregated from transaction-level data and lag-based features representing the previous three months. Training employed `TimeSeriesSplit` [3] [4] with five folds, and performance was evaluated using Mean Absolute Error (MAE). The Expense Forecast Model was implemented using `RidgeCV` [14] to incorporate ℓ_2 regularization, with features including lagged expenses, rolling statistics, and seasonal indicators. The regularization parameter α was selected from a logarithmic range (10^{-3}–10^{3}) via time-aware cross-validation.

Category-level budget prediction was implemented using `GradientBoostingRegressor` [10] [11] to capture nonlinear spending behavior, with engineered temporal and lag-based features. Key hyperparameters included 400 estimators, a learning rate of 0.05, and a maximum tree depth of 3. The Coupon Recommendation Model employed a hybrid TF-IDF [5] [6] and k-Nearest Neighbors [7] [6] similarity framework, using cosine similarity with $k = 5$ and rule-based filtering to refine recommendations.

4.3 Results and Discussion

The experimental results demonstrate strong overall performance across all proposed models. For income prediction, the proposed Ridge model achieved 2.84% MAPE, performing comparably to the naïve last-month baseline (2.86%) and slightly underperforming the 3-month moving average baseline (2.43%). This suggests that income exhibits strong temporal stability and can be effectively modeled using simple smoothing techniques. In contrast, the expense forecasting model achieved 22.8% MAPE, substantially outperforming the naïve baseline (40.0%) and the 3-month moving average (37.0%), reducing absolute percentage error by approximately 43% relative to the naïve approach. Despite negative R^2 values caused by high variance and temporal volatility, the proposed model consistently minimizes absolute forecasting error. For category-level budget prediction, the model achieved 24.6% MAPE compared to 31.9% (naïve) and 27.0% (MA3), demonstrating improved error reduction across categories; however, negative R^2 values indicate substantial heterogeneity and low signal-to-noise ratio at fine-grained category levels. Finally, the Coupon Recommendation Model, built using TF-IDF and k-Nearest Neighbors, achieved 95% semantic similarity accuracy

[1] A GitHub repository has been developed for the model (`https://github.com/mallikarjunayeruku/Personal-Finance-Management-using-Artificial-Intelligence-Models`). Access can be granted upon request.

TABLE 2. Forecasting Performance Comparison

	Metric	Proposed	Last	MA(3)
Income Model	MAPE (%)	2.84	2.86	**2.43**
	R^2	—	—	—
Expenses Model	MAPE (%)	**22.8**	40.0	37.0
	R^2	-10.13	-27.49	-17.51
Category Budget	MAPE (%)	**24.6**	31.9	27.0
	R^2	-1.15	-1.03	-0.49

Income Prediction Model : 8-fold rolling-origin CV. Expenses Prediction Model: 6-fold rolling-origin CV (1,557 transactions). Category Budget Prediction Model: rolling-origin CV across categories. Baselines: Last Month and 3-Month Moving Average.

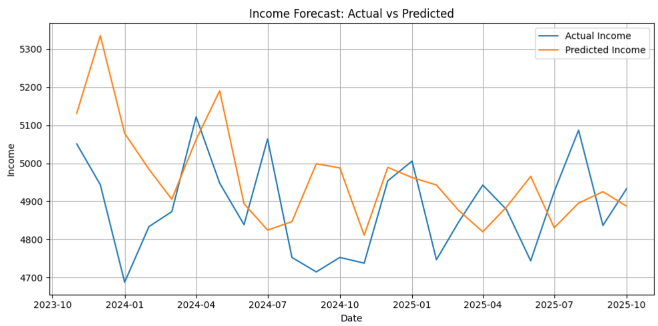

FIGURE 3. Income Prediction Model

in relevant product matching while filtering irrelevant items, delivering estimated savings of 8–15% per item and confirming practical real-world cost-optimization capability.

As observed in Figure 3 and Figure 4, the larger deviations in the initial months result from limited lag history and smaller training windows in early rolling-origin folds. With reduced temporal context, the regularized regression model operates in a higher-bias regime, producing smoother predictions that underreact to abrupt fluctuations. As the training window expands, the bias–variance balance improves and forecast stability increases.

5. Conclusions and Future Work

This paper proposes a scalable machine learning framework for intelligent personal financial decision support that goes beyond conventional transaction tracking by integrating predictive analytics with personalized, actionable insights. Empirical evaluation under time-aware validation demonstrates strong predictive accuracy: the income forecasting model attains low

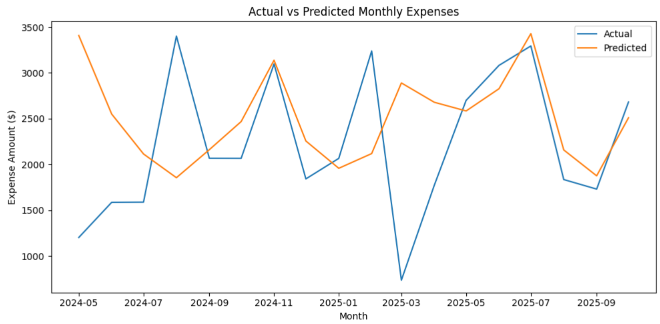

FIGURE 4. Expenses Prediction Model

absolute error with a relative deviation on the order of only a few percent, while the expense forecasting model achieves moderate absolute error consistent with real-world spending variability. These results support the effectiveness of the proposed approach even under privacy-preserving data constraints that limit feature availability and data granularity.

Future work will prioritize evaluation on larger, real-world financial datasets, broader multi-source data integration, and richer feature representations leveraging advanced NLP techniques. From a modeling perspective, we will investigate deep learning architectures such as LSTM [12], GRU, and Temporal Fusion Transformers, as well as gradient-boosted ensemble methods including XGBoost [16] and LightGBM [17], to better capture non-linearities and long-range temporal dependencies in personal finance behavior. We also plan to explore hierarchical forecasting and adaptive budget optimization to improve personalization, stability, and robustness across heterogeneous user profiles.

References

[1] S. VADUKA, S. C. REDDY, H. V. ARIKATHOTA, L. KONCHADA, P. CHAKRABORTY, and A. BANDYOPADHYAY, "Optimizing personal finance management through ai-driven decision support systems," in *2024 IEEE Region 10 Symposium (TENSYMP)*, 2024, pp. 1–6.

[2] A. Verma, C. Kapoor, A. Sharma, and B. Mishra, "Web application implementation with machine learning," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 2021, pp. 423–428.

[3] H. Ni, "Topology regressive distributed model for financial time series prediction," in *2009 Fifth International Conference on Natural Computation*, vol. 3, 2009, pp. 463–467.

[4] K. Gupta, D. K. Tayal, and A. Jain, “An experimental analysis of state-of-the-art time series prediction models,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022, pp. 44–47.

[5] Q. Liu, J. Wang, D. Zhang, Y. Yang, and N. Wang, “Text features extraction based on tf-idf associating semantic,” in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 2338–2343.

[6] L. Li, B. Zhang, Y. Che, and M. Yang, “Kd-knn text categorization method based on improvement tfi-df,” in *2009 International Conference on Information Engineering and Computer Science*, 2009, pp. 1–5.

[7] T. Li, “On knn and svm text classification technology in knowledge management,” in *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*, vol. 8, 2011, pp. 3923–3926.

[8] “Exploring the role of digital financial tools in personal finance: Enhancing financial planning and decision-making,” in *2024 International Conference on Computer and Applications (ICCA)*, 2024, pp. 1–6.

[9] S. G. Kaushik, S. S. Kashyap, S. B. V, C. M, S. Reddy, and A. N, “Personal finance management and prediction using ml algorithms,” in *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 2024, pp. 1–5.

[10] P. Bahad and P. Saxena, “Study of adaboost and gradient boosting algorithms for predictive analytics,” 01 2020, pp. 235–244.

[11] M. Chi, “Optimization of prediction model based on gradient boosting decision tree,” in *2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 2024, pp. 1–5.

[12] C. Fjellström, “Long short-term memory neural network for financial time series,” in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 3496–3504.

[13] C.-J. Wu, W.-S. Zeng, and J.-M. Ho, “Optimal segmented linear regression for financial time series segmentation,” in *2021 International Conference on Data Mining Workshops (ICDMW)*, 2021, pp. 623–630.

[14] M. S. K. Reddy, R. Sumathi, N. V. K. Reddy, N. Revanth, and S. Bhavani, “Analysis of various regressions for stock data prediction,” in *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 2022, pp. 538–542.

[15] X. Pang and Y. Liao, “A text classification model based on training sample selection and feature weight adjustement,” in *2010 2nd International Conference on Advanced Computer Control*, vol. 3, 2010, pp. 294–297.

[16] L. Jidong and Z. Ran, “Dynamic weighting multi factor stock selection strategy based on xgboost machine learning algorithm,” in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, 2018, pp. 868–872.

[17] M. R. Machado, S. Karray, and I. T. de Sousa, “Lightgbm: an effective decision tree gradient boosting method to predict customer loyalty in the finance industry,” in *2019 14th International Conference on Computer Science Education (ICCSE)*, 2019, pp. 1111–1116.