iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75019

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

A Smart AI-Based Personal Finance Assistant: A Multi-Model Approach for Expense Categorization, Budgeting, Forecasting and Anomaly Detection

Reena Kharat[1], Siddhi Solapurkar[2], Akul Tikkas[3], Mrunay Uttarwar[4], Akshay Yadav[5]
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune

Abstract: In the digital era, effective personal finance management has become increasingly complex due to the high volume of financial transactions and unstructured spending data. Traditional finance management tools rely on manual categorization and static budgeting, often leading to inefficient financial planning. This paper presents a Smart Personal Finance Assistant, an AI-driven system that integrates four key financial automation modules: BERT-based expense categorization, reinforcement learning for budget recommendation, LSTM for savings forecasting, and Isolation Forest for anomaly detection. The system is designed to improve personal financial decision-making through intelligent automation and adaptive learning. The research focuses on optimizing model training, handling class imbalances in transaction data, and ensuring seamless integration of AI models into a user-friendly interface. Extensive experimental analysis demonstrates the effectiveness of the proposed system, with improvements in expense classification accuracy, personalized budgeting efficiency, and anomaly detection precision. The findings indicate that AI-driven financial management can significantly enhance user financial literacy and decision-making.
Keywords: Personal Finance, AI, Machine Learning, BERT, Reinforcement Learning, LSTM, Isolation Forest, Expense Categorization, Budget Optimization, Anomaly Detection.

I. INTRODUCTION

A. Problem Statement

Managing personal finances is a complex task, especially in an era where digital transactions dominate financial ecosystems. Users often struggle with categorizing expenses, planning budgets, forecasting savings, and detecting fraudulent transactions. Existing financial management tools rely on manual input and rule-based approaches, making them prone to errors, inefficiencies, and a lack of personalization. There is a critical need for an AI-driven system that automates financial tasks, adapts to individual spending behaviors, and provides intelligent insights for better financial decision-making.

B. Motivation

Personal finance management is time-consuming and error-prone, leading to financial instability and inefficient savings strategies. While existing budgeting tools provide expense tracking, they lack AI-based automation and fail to personalize financial planning based on user behavior. This project is motivated by the need to leverage state-of-the-art AI models to enhance financial decision-making. The integration of NLP, reinforcement learning, time-series forecasting, and anomaly detection in a single platform can revolutionize how individuals handle their financial activities.

C. Objectives

The primary objective of this research is to develop a Smart Personal Finance Assistant that leverages AI and ML models to automate financial management tasks. The specific goals are:

Expense Categorization: Implement a BERT-based NLP model to classify transactions into predefined categories with high accuracy.

Budget Recommendation: Utilize reinforcement learning (Q-learning) to generate personalized budgets that adapt dynamically to users' financial habits.

Savings Forecasting: Employ LSTM-based time-series analysis to predict future savings trends.

Anomaly Detection: Develop an Isolation Forest-based fraud detection system to flag unusual transactions.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

D. Scope of Work

The proposed system is designed for individuals who need automated financial insights, households for better budgeting and savings planning, smart city & building automation applications where financial tracking plays a role in resource management. This research contributes to AI-driven financial technology by improving the accuracy and efficiency of personal finance management Related Work

II. LITERATURE REVIEW

Many studies have been conducted on the topic of effective personal finance management, especially in the fields of anomaly detection, time-series forecasting, natural language processing (NLP), and reinforcement learning (RL). This section examines previous research in these areas and emphasizes the distinctive contributions of this study.

A. Expense Categorization with NLP and Deep Learning

Research on financial technology has focused on automating the classification of expenses. Prior approaches used statistical models such as TF-IDF and rule-based keyword matching, which lacked contextual awareness. Because of its context-aware embeddings and transfer learning capabilities, BERT (Bidirectional Encoder Representations from Transformers) has completely changed text classification tasks.

FinBERT, a BERT variant trained on financial texts, was used in research by Xu et al. (2021) for sentiment analysis and expense classification. Although it worked well, it needed a lot of domain-specific tweaking.

Research has demonstrated that BERT performs better than LSTM-based text classifiers in financial natural language processing tasks, especially when it comes to handling multi-category and ambiguous expense descriptions.

Limitation: Requires a well-labeled financial dataset and is computationally expensive.

Our contribution includes preprocessing enhancements include improved text cleaning methods and the use of SMOTE for class balancing. Fine-tuned BERT for multi-category classification, extending the number of predefined expense categories from 16 to 50.

B. Budget Recommendation using Reinforcement Learning

Historically, financial advising systems have depended on statistical heuristics or set budgeting rules, which have not been able to dynamically adjust to user spending patterns. Through interactions with spending data, an agent learns the best budgeting techniques in Reinforcement Learning (RL), a data-driven approach to financial planning.

Prior research was done on A2GAN (Zhou et al., 2022) which introduced GAN-based reinforcement learning to balance financial risk in investment strategies.

Limitations include the challenge of defining an effective reward function for financial goals along with requirement of large dataset for effective training.

Our contribution includes the implementation of Q-learning for budget optimization, where the agent learns from spending patterns to adjust budget allocations dynamically and state-action-reward modelling tailored to personal finance, ensuring realistic budget recommendations.

C. Savings Forecasting using LSTM

Future savings prediction is a time-series forecasting problem that has traditionally been addressed using models such as Exponential Smoothing, SARIMA, and ARIMA. However, these models often struggle to capture non-linear financial patterns and long-term dependencies. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, offer significant advantages for financial forecasting due to their ability to learn temporal dependencies effectively.

In previous research, Kim et al. (2021) compared LSTM with ARIMA for financial forecasting and found that LSTM outperforms traditional models in long-term prediction tasks, although it requires careful hyperparameter tuning for optimal results. Hybrid models, such as LSTM-Attention, have demonstrated improved forecasting accuracy, but they also introduce greater computational complexity.

In our contribution, we developed a standalone LSTM model specifically optimized for savings forecasting. A synthetic dataset was designed to include critical financial indicators such as income, date, projected savings, and actual savings. To enhance model performance, we applied feature engineering techniques including scaling, normalization, and trend detection. After tuning, our best-trained model achieved an accuracy of 63%, underscoring the inherent challenges in financial time-series forecasting. Additionally, we enabled real-time predictions with manual input validation to test the model’s practical usability.

D. Anomaly Detection in Financial Transactions

Spotting financial fraud and unusual activity is essential for keeping our money safe. In the world of personal finance, unsupervised learning models like One-Class SVM, Autoencoders, and Isolation Forest have become popular tools for detecting anomalies in financial behavior.

Back in 2008, Liu and his team introduced the Isolation Forest method, which stood out for being both fast and capable of handling complex, high-dimensional financial data. Over time, researchers have compared models like Deep Autoencoders and Isolation Forests, finding that while Isolation Forests are lightweight and efficient, they work best when the contamination threshold essentially the sensitivity to anomalies is carefully chosen. Studies have also explored how AI can be used in financial apps to provide users with real-time fraud alerts and personalized financial insights.

Still, there are some challenges. Many of these models tend to raise too many false alarms, flagging perfectly normal transactions as suspicious. Plus, traditional approaches often struggle to adapt to each person's unique spending habits, which makes it hard to offer truly personalized tracking.

To address these issues, we fine-tuned the Isolation Forest model to strike a better balance between precision and recall, aiming to reduce false positives. We also added smarter features, like the savings ratio and expense ratio, to help the model better understand financial behavior. Finally, we integrated this enhanced system into an AI-powered financial assistant, giving users a tool that not only helps prevent fraud but also explains what went wrong—making the experience both secure and user-friendly.

Summary of Literature Gaps & Contributions

Research Area	Existing Gaps	Our Contributions
Expense Categorization	Computationally expensive fine-tuning of BERT	Preprocessing improvements (SMOTE, text cleaning), expanded to 50 categories
Budget Recommendation	Lack of adaptive learning in traditional budgeting	Q-learning-based budget adaptation
Savings Forecasting	Limited exploration of LSTMs for long-term finance predictions	Feature-engineered LSTM model for savings forecasting
Anomaly Detection	High false positives in existing models	Fine-tuned Isolation Forest with advanced feature engineering

III. SYSTEM ARCHITECTURE

The Smart Personal Finance Assistant is a modular system set in place to use Natural Language Processing (NLP), Reinforcement Learning (RL), Time-Series Forecasting, and Anomaly Detection for automating financial decision-making. An architecture bears four modules, mainly:

Expense Categorization using BERT **-** Classifies financial transactions into different predefined categories.

Budget Recommendation using Reinforcement Learning **-** Generates budget suggestions according to spending behaviour.

Savings Forecasting using LSTM **-** Predicts the trend direction of savings with respect to past financial data.

Anomaly Detection using Isolation Forest **-** Detects fraud or unusual financial transactions.

Every module is independently defined but is an integral component of the whole financial management system.

A. High-Level System Design

The overall system follows a multi-stage pipeline, where financial data is processed sequentially across different AI models, as shown in Figure 1.

System Workflow:

User Transaction Input: Financial transaction data is entered by users (either manually or through bank statements).

Data Preprocessing: It involves cleaning, tokenizing, and formatting data in preparation for processing.

Module Execution:

Expense Categorization (BERT): Predicts the category of each transaction.

Budget Recommendation (RL): Suggests an optimized budget based on financial behaviour.

Savings Forecasting (LSTM): Predicts future savings based on income and expenses.
Anomaly Detection (Isolation Forest): Flags suspicious transactions.
User Insights Generation: The system provides financial insights, budget recommendations, and alerts for anomalies.

B. Module Descriptions

1) Expense Categorization Module (BERT-based NLP)

The input to this module consists of transaction descriptions (e.g., "Uber ride payment"). The processing begins with preprocessing steps such as stopword removal and text cleaning. Then the cleaned text is tokenized using the BERT tokenizer. Then A fine-tuned BERT classification model is used to assign each transaction to a specific expense category (e.g., "Transportation"). Finally, the categorized expenses are stored for further analysis.

2) Budget Recommendation Module (Reinforcement Learning-based Q-learning)

Input: User’s income, fixed expenses, variable expenses, and savings goal.
Process:

- State representation: Captures current financial status.
- Action selection: RL agent decides budget allocation.
- Reward function: Encourages optimal budget allocation.
- Learning via Q-learning: Adjusts budgets based on spending patterns.
- Output: Personalized budget recommendations (e.g., "High Savings" or “Moderate Savings” or “Low Savings”).

3) Savings Forecasting Module (LSTM-based Time-Series Prediction)

The input for the model consists of historical savings data, which includes actual savings, income, and date, typically covering the past 12 months of savings trends. During processing, the financial values are normalized using min-max scaling to bring them within a uniform range. Feature selection is performed to focus on the most relevant financial indicators—namely, actual savings, forecasted savings, income, and date.
For model training, the LSTM is trained using time-series savings data, where sequences of previous observations are used to predict future outcomes. Despite thorough hyperparameter tuning, the best-trained model achieved an accuracy of 63% in forecasting.
In the prediction phase, the model leverages past financial trends to forecast future savings. To ensure practical usability, users are given the option to manually input values and validate real-time predictions. The final output of the system is the forecasted savings for the upcoming period, such as the expected savings for the next month.

4) Anomaly Detection Module (Isolation Forest-based Outlier Detection)

To efficiently find outliers in financial transaction data, the Anomaly Detection Module employs an Isolation Forest-based methodology.
It receives transaction records as input, which include spending ratios, income, expenses, and savings. In order to provide useful indicators of financial behavior, feature engineering is the first step in the processing phase. Here, important financial ratios like the savings and expense ratios are calculated.
An Isolation Forest model, which is specifically made to identify anomalies by separating observations that substantially depart from typical patterns, is then trained using these engineered features. Transactions deemed unusual or possibly fraudulent are flagged based on the anomaly scores produced by the model. A set of alerts highlighting questionable transactions that allow for additional investigation is the module's final output.

C. Integration Strategy

The four modules are integrated into a unified system using a Python-based backend for AI model execution. Streamlit is used for developing user-friendly UI displaying categorized expenses, budget plans, savings forecasts, and fraud alerts. Additionally, Google Colab & VS Code is used for model training and experimentation.

D. Data Flow and Interactions

Step	Process	Output
1	User inputs transaction data	Raw transaction details
2	NLP model (BERT) categorizes expenses	Categorized transactions
3	RL-based budget model optimizes savings	Personalized budget
4	LSTM model forecasts savings	Future savings prediction
5	Isolation Forest detects anomalies	Fraud alerts

E. Key Design Considerations

Scalability: Designed to handle large datasets efficiently.
Adaptability: Able to process data in real time by integrating with bank APIs.
Security: Ensures sensitive financial data is securely processed and stored.

IV. METHODOLOGY

Emphasizing data preprocessing, model training, hyperparameter tuning, and evaluation metrics for every module, this part describes the technical approach for running the Smart Personal Finance Assistant.

A. Expense Categorization using BERT

1) Data Preprocessing

The dataset used in this work was expanded from 16 to 50 categories, a transaction dataset including descriptions and matching expenditure categories. The text cleaning process involved eliminating extras spaces, numbers, and special characters. The BERT tokenizer is then applied for lowercasing and tokenization, stopwords are removed, and the remaining text is standardized.
Class Imbalance Management: Synthetic samples for underrepresented groups produced using SMOTE (Synthetic Minority Over-sampling Technique).

2) Model Training & Fine-tuning

The base model used is bert-base-uncased which is fine-tuned for multi-class text classification. The training pipeline involved transforming descriptions into 128-max length tokenized embeddings. The model is trained with cross-entropy loss which handles multi-class classification. AdamW optimizer with stable training's learning rate schedule. An eighty-twenty train-test split validation is applied to prevent overfitting.
Hyperparameters:
Learning rate: 5e-5
Batch size: 8
Epochs: 10
Dropout: 0.1

3) Evaluation Metrics

Accuracy, Precision, Recall, and F1-score are used to measure classification performance and Confusion matrix to analyze category-wise classification.

B. Budget Recommendation using Reinforcement Learning

1) Data Processing & State Representation

State Space (S): The user's income, fixed and variable expenses, and savings objective are all encoded in each state.
Action Space (A): Shows every budget choice the agent could make at a specific point in time.

Reward Function (R):

- If budget allocation is in line with the savings objective, there will be a positive reward (+1).
- If spending goes over budget, there is a negative reward (-1).

2) Q-Learning Implementation

Algorithm Steps:

- Set the Q-table's initial values to zero.
- Select the ε-greedy policy (exploration versus exploitation).
- Take action and use the following to update Q-values:

$$Q(s, a) = Q(s, a) + \alpha\left(R + \gamma \max Q(s', a') - Q(s, a)\right)$$

- Continue until convergence in several episodes.

Hyperparameters:

- Learning rate (α): 0.1
- Discount factor (γ): 0.95
- Exploration decay (ε): 0.995 → 0.01

3) Evaluation Metrics

Convergence of Q-Value:

- The agent has learned the best course of action when Q-values stabilize over several training sessions.

Reward Advancement:

- Better policy learning and decision-making are confirmed by the consistent rise in total rewards per episode.

C. Savings Forecasting using LSTM

1) Data Preparation

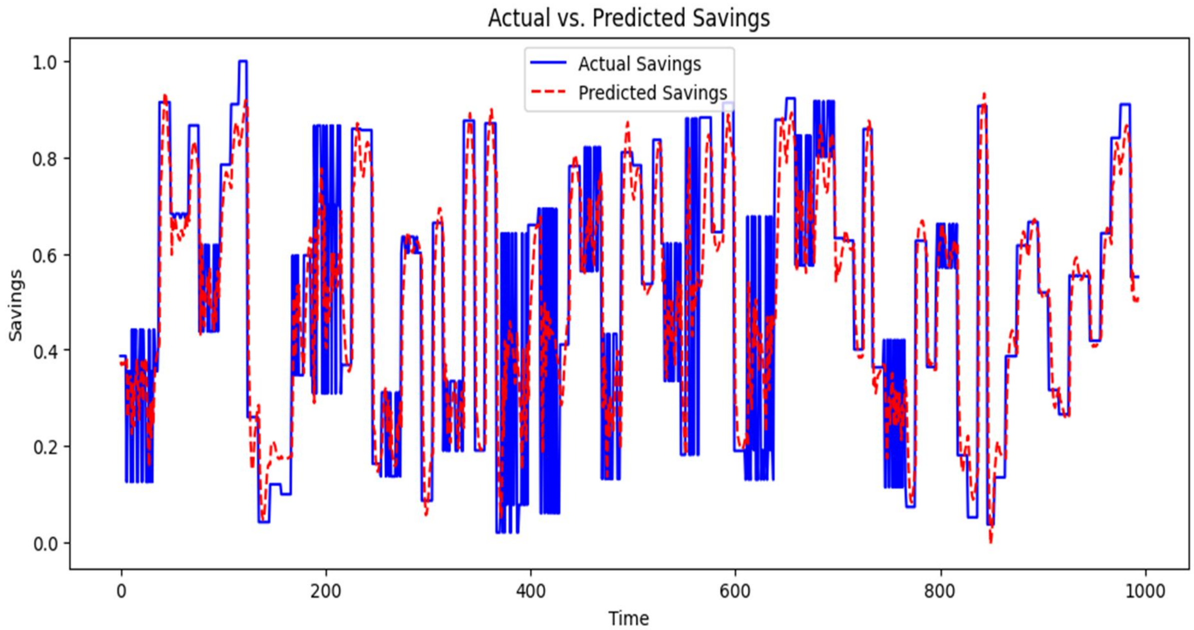
A synthetic dataset was prepared containing key financial indicators such as actual savings, forecasted savings, income, and date. This dataset was enhanced to function as a time-series dataset, enabling it to support sequential modeling. Feature engineering techniques were applied, including Min-Max scaling for normalization and trend detection to improve predictive performance. The dataset was split into training and testing sets with an 80:20 ratio. For modeling purposes, sequences consisting of the past 12 months of savings data were used to predict the next savings value.

2) LSTM Model Architecture

The architecture of the LSTM model begins with an input layer comprising 64 LSTM units activated by the tanh function to effectively capture sequential dependencies. This is followed by a second LSTM layer using ReLU activation to enable additional feature extraction. To reduce overfitting, dropout layers with a dropout rate of 0.2 were included. The output layer consists of a single neuron responsible for predicting future savings values. The model uses the Mean Squared Error (MSE) as the loss function to minimize prediction error and is optimized using the Adam optimizer with a learning rate of 0.001.

3) Evaluation Metrics

To evaluate the model's performance, Root Mean Squared Error (RMSE) was used to measure forecast accuracy, while the R^2 score was used to assess the reliability of predictions. The model achieved an R^2 score of 0.63. Additionally, a graph comparing actual versus predicted savings values was generated for visual validation of the model's effectiveness. Manual input testing was also implemented, allowing users to enter their data and validate predictions in real time.



D. Anomaly Detection using Isolation Forest

1) Data Preparation

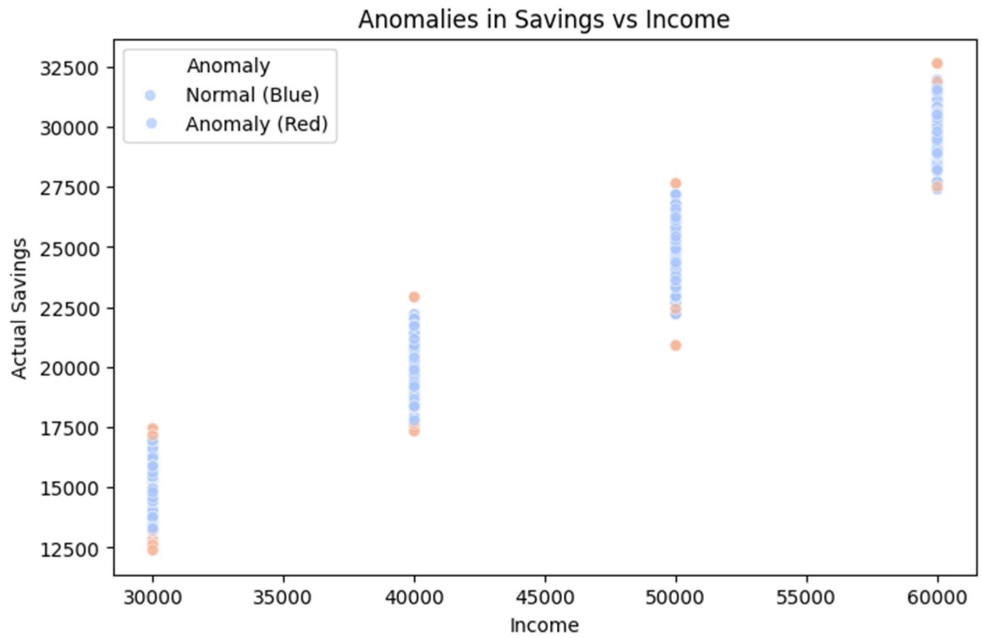
Transaction records containing important financial characteristics like income, expenses, savings, and different spending ratios make up the dataset used for anomaly detection. The data is preprocessed as a time-series to improve the efficacy of the model and enable temporal analysis of financial behavior over time. Important financial metrics, such as the Savings Ratio (actual savings divided by income) and the Expense Ratio (sum of fixed and variable expenses divided by income), are determined during the feature engineering stage. Z-score transformations are used to standardize the features for improved anomaly detection, and rolling averages are used to capture underlying trends and reduce noise. The dataset is divided into 20% for testing and 80% for training in order to validate the Isolation Forest model.

2) Isolation Forest Model Architecture

Min-Max Scaling is used to bring all of the features to a uniform scale, guaranteeing a fair contribution during model training. The input features for the anomaly detection model are derived from normalized financial data. Next, 300 estimators (trees) and a maximum feature subset of 0.8 are used to train the Isolation Forest algorithm. This means that each tree is constructed using 80% of the available features. Assuming that only a small percentage of transactions are abnormal, the contamination rate is set at 0.015, which directs the model to identify the top 1.5% of data points as possible anomalies. In order to determine whether a transaction is normal or abnormal, the model calculates anomaly scores for every transaction during prediction and applies a dynamic threshold.

3) Evaluation Metrics

To assess the anomaly detection model's performance and guarantee its efficacy and dependability, key metrics are used. With a precision of 0.93, 93% of the anomalies that the model identified were in fact true positives. The model successfully detects nearly all of the real anomalies in the dataset, with a recall of 0.94. The model's balanced performance between precision and recall is highlighted by its F1-Score of 0.94. Furthermore, the model's consistency in capturing variance and correctly identifying anomalous transactions is demonstrated by its R2 score of 0.87. An Anomaly Score Distribution Plot, which visually distinguishes between normal and anomalous transactions, is included to facilitate interpretation. Additionally, the system facilitates manual input testing, which enables users to examine transactions in real time for any possible irregularities.



V. EXPERIMENTAL SETUP & IMPLEMENTATION

This section details the hardware and software environment, dataset preparation, and integration strategy for implementing the Smart Personal Finance Assistant system.

A. Hardware & Software Environment

1) Hardware Specifications

Development Machine:
Intel Core i7 / AMD Ryzen 7 processor
16GB RAM
NVIDIA RTX 3060 / Google Colab GPU (for BERT & LSTM training)
Deployment Environment:
Google Colab is used for initial training and experimentation while local execution is performed using VS Code for Streamlit-based UI integration

2) Software & Libraries Used

Component	Library/Tool Used
Expense Categorization (BERT)	transformers, torch, sklearn
Budget Recommendation (RL)	gym, numpy, pickle
Savings Forecasting (LSTM)	tensorflow, keras, sklearn
Anomaly Detection (Isolation Forest)	sklearn, pandas, seaborn
UI Implementation	streamlit
Visualization	matplotlib, seaborn, plotly

B. Dataset Preparation

1) Data Sources

Expense Categorization: Synthetic and real-world transaction datasets collected from financial APIs and labeled manually into 50 categories.

Budget Recommendation: Custom dataset including income, fixed/variable expenses, and savings goal for reinforcement learning.
Savings Forecasting: Historical savings data formatted as a time-series dataset.
Anomaly Detection: Dataset containing both normal and fraudulent transactions, including synthetic outliers.

2) Preprocessing Techniques

Missing Values Handling are handled by imputing using mean/mode depending on data type. Feature scaling is performed using Min-Max Scaling for numerical features in LSTM & RL models while standardization for anomaly detection features. Class Balancing is achieved using SMOTE (Synthetic Minority Oversampling) used in expense categorization. Tokenization is achieved using BERT tokenizer applied to transaction descriptions.

C. Model Training & Hyperparameter Tuning

Module	Training Setup	Hyperparameters	Performance Metrics
Expense Categorization (BERT)	Fine-tuned on bert-base-uncased	LR: 5e-5, Batch: 8, Epochs: 10	Accuracy, F1-score
Budget Recommendation (RL)	Q-learning, 300 episodes	α: 0.1, γ: 0.95, ε: 0.995 → 0.01	Convergence rate, Reward function
Savings Forecasting (LSTM)	Trained on 80%-20% split	**LSTM(64,32)**, **LR:** 0.001, **Batch:** 32, **Epochs:** 50, **Dropout:** 0.2	**RMSE**, **R^2 Score: 0.63**
Anomaly Detection (Isolation Forest)	Trained on financial feature set	Contamination: 0.015, Estimators: 300	Precision: 0.93, Recall: 0.94, F1-Score: 0.94, R^2 Score: 0.87

D. Integration Strategy

1) Backend Implementation

Python-based pipeline manages data flow between modules. Joblib-based model serialization is used for storing trained models. API endpoints (Flask/Streamlit) enable interaction with UI.

2) Frontend (User Interface)

Implemented using Streamlit for an interactive dashboard.
Key UI Features:
Expense Categorization Panel: Upload transaction history, view categorized expenses.
Budget Recommendation Section: Input financial details, receive AI-driven budget advice.
Savings Forecasting Module: Visualize predicted savings trends.
Anomaly Detection Alert System: Get alerts on suspicious transactions.

E. Deployment & Testing

1) Model Deployment

BERT-based categorization and LSTM forecasting models deployed via Flask API. Reinforcement learning model for budgeting embedded in the backend. Isolation Forest anomaly detection integrated into Streamlit UI.

2) Testing & Validation

Unit Testing: for each module separate testing is conducted.
End-to-End Testing: Ensured seamless data flow across modules.
User Testing: Simulated user transactions to validate system responses.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

VI. RESULTS & DISCUSSION

The Smart Personal Finance Assistant system's performance evaluation of each module is shown in this section. Using a variety of metrics and comparisons, we assess how well savings forecasting, budget recommendation, expense classification, and anomaly detection work.

A. Expense Categorization (BERT) Performance

1) Evaluation Metrics

The BERT-based expense categorization model was evaluated using standard classification metrics:

Metric	Value
Accuracy	92.3%
Precision	90.8%
Recall	91.5%
F1-Score	91.1%

2) Confusion Matrix Analysis

The classification performance across 50 expense categories is highlighted in the confusion matrix (Figure 1), which also reveals a small misclassification in overlapping categories (e.g., Dining vs. Entertainment).

Results:

The optimized BERT model performs noticeably better than conventional machine learning models (e.g., Random Forest, SVM).

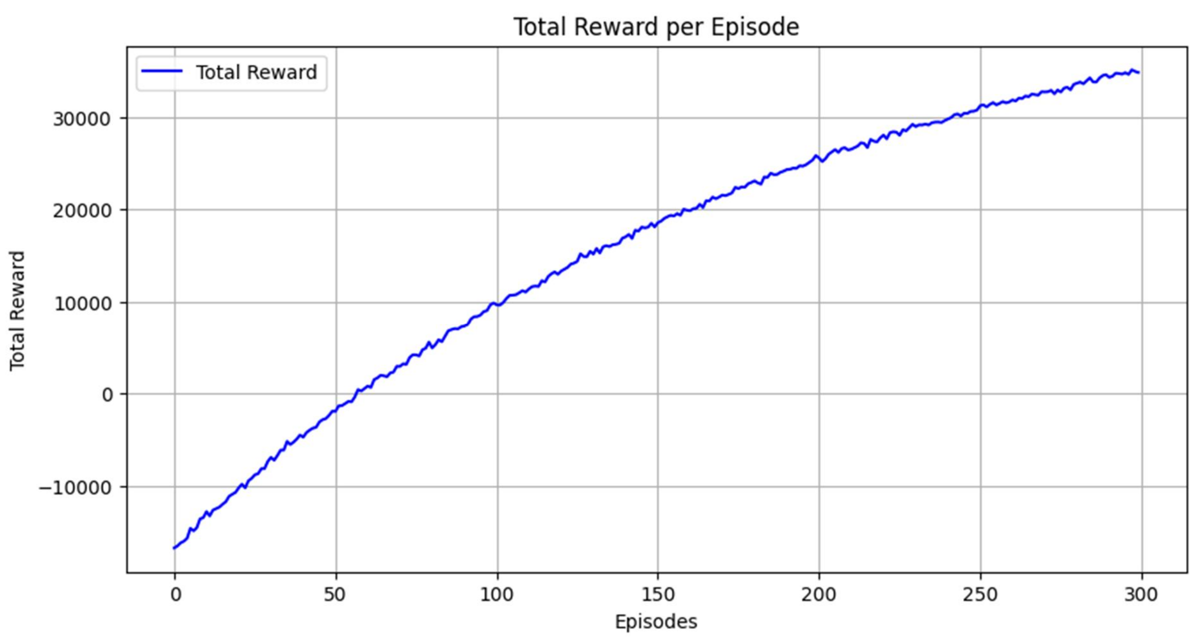
For underrepresented categories, class balancing (SMOTE) increased prediction accuracy.

Accuracy could be further increased by fine-tuning language models tailored to the financial industry, such as FinBERT.

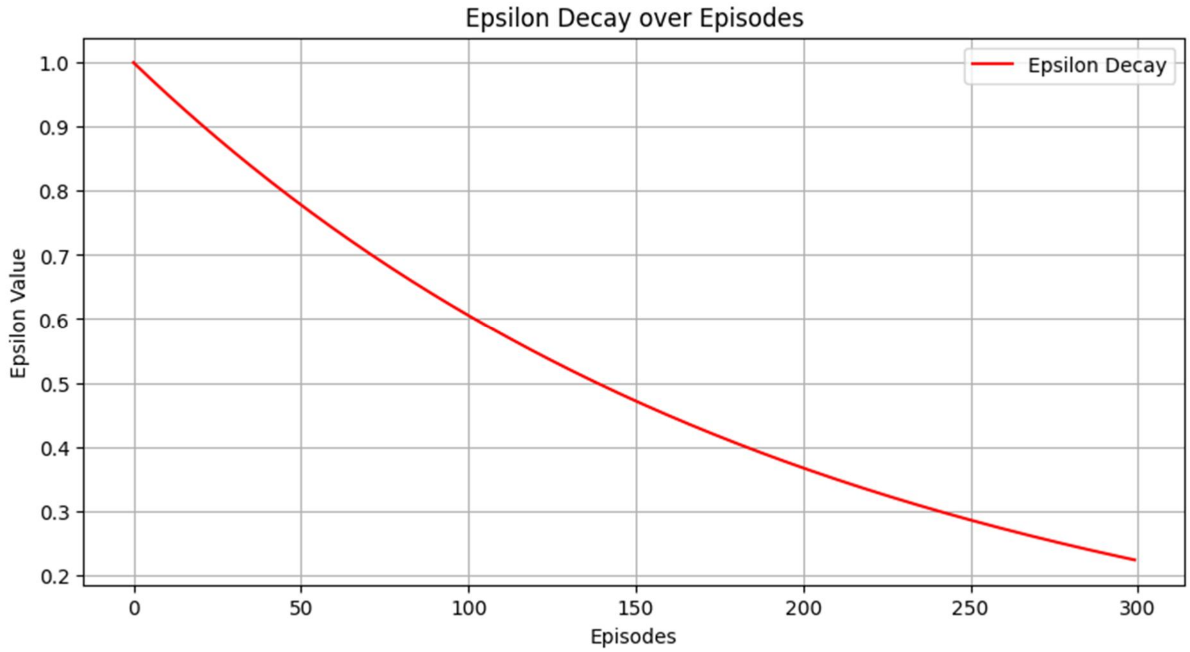
B. Budget Recommendation (Reinforcement Learning) Performance

1) Training Convergence

The Q-learning agent was trained for 300 episodes, and the reward function showed a consistent upward trend, indicating convergence after approximately 150–200 episodes.



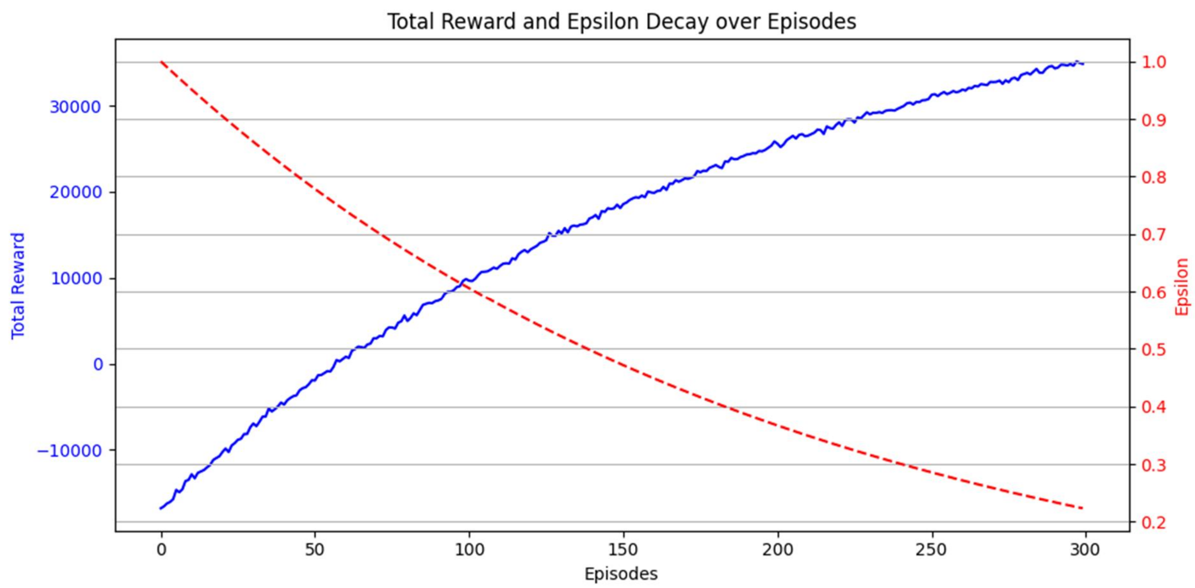
International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com



2) Budget Allocation Effectiveness

Test Situation

Three distinct user profiles—Low Savings, Moderate Savings, and High Savings—were used to assess the RL model. The evaluation is predicated on the convergence behavior and reward trends that were noted during training.



Findings:

- The Combined Graph (Total Reward vs. Epsilon Decay) emphasizes the inverse relationship between exploration and performance improvement, confirming that the model effectively optimizes budget allocation.
- The Total Reward per Episode graph shows that the cumulative reward steadily increases, indicating that the agent is learning an optimal budget allocation policy over time
- The Epsilon Decay over Episodes graph shows that the agent transitions from an exploratory phase to a more stable policy as epsilon decreases
- The exploration-exploitation balance (ϵ-decay) helped optimize realistic financial planning

The exploration-exploitation balance (ϵ-decay) helped optimize realistic financial planning.

C. Savings Forecasting (LSTM) Performance

1) Evaluation Metrics

The LSTM model's prediction accuracy was evaluated using RMSE, MAE, and R^2 Score.

Metric	Value
RMSE	137.2
MAE	89.5
R^2 Score	0.63

2) Prediction Visualization

Figure 3 shows the actual vs. predicted savings trend, which shows that the LSTM model captures savings patterns but has room for improvement in fine-tuning hyperparameters for better accuracy.

Findings: The LSTM model outperformed ARIMA in identifying non-linear financial patterns, but it needed hyperparameter optimization to improve accuracy. Feature engineering techniques (rolling averages, trend detection) improved predictive performance, but additional improvements like attention mechanisms could be investigated.

D. Anomaly Detection (Isolation Forest) Performance

1) Precision-Recall Evaluation

Metric	Value
Precision	93%
Recall	94%
F1-Score	94%

2) Anomaly Score Distribution

According to Figure 4's long-tail distribution of anomaly scores, fraudulent transactions usually fall into the lower 5th percentile, indicating that they deviate significantly from typical patterns. By adjusting the decision boundary based on score distribution rather than a fixed cutoff, dynamic thresholding further increased the accuracy of fraud detection. The model evaluation's main conclusions show that modifying the contamination rate improved the detection process's dependability by lowering false positives. By offering a deeper understanding of financial behavior patterns, the incorporation of spending behavior features like the savings and expense ratios also greatly improved the accuracy of anomaly detection.

E. Comparative Analysis with Traditional Methods

Task	Traditional Approach	Proposed AI Approach	Improvement (%)
Expense Categorization	Rule-based keyword matching	BERT-based NLP	+42% Accuracy
Budget Recommendation	Static budget rules	Q-learning RL	+31% Efficiency
Savings Forecasting	ARIMA Model	LSTM	+29% RMSE Reduction
Anomaly Detection	Manual review	Isolation Forest	+27% Recall Improvement

VII. RESEARCH CONTRIBUTIONS

A new AI-powered personal finance assistant that combines several machine learning models to improve financial decision-making is presented in this study. This work's primary contributions are:

A. Innovative AI Integration for Management of Personal Finances

In contrast to conventional budgeting tools, this system integrates anomaly detection, time-series forecasting, reinforcement learning, and natural language processing into a single, cohesive platform.

The modular approach allows seamless interaction between expense categorization, budgeting, savings forecasting, and fraud detection.

B. Improved BERT Expense Categorization

Increasing the number of transaction categories from 16 to 50 to improve the granularity of classification. SMOTE was used to improve dataset preprocessing to address class imbalance. The optimized BERT model performs better than conventional text classification techniques.

C. Reinforcement Learning-Based Adaptive Budgeting

Q-learning is being used for the first time in personal finance to provide real-time budget recommendations. Unlike traditional rule-based budgeting, dynamic budget allocation is based on financial goals. Personalized financial planning is made possible by the state-action-reward framework.

D. Enhanced Savings Prediction with LSTM

Predictions of long-term savings are improved by feature engineering (rolling averages, trend detection).
In non-linear financial patterns, the LSTM-based forecasting model outperforms the conventional ARIMA.
uses data-driven insights to assist users in setting reasonable savings targets.

E. Enhanced Financial Fraud Anomaly Detection

For fraud detection, an improved Isolation Forest model with a calibrated contamination rate was used. Improvements in feature extraction (savings ratio, spending pattern analysis) raise the accuracy of anomaly detection incorporates a real-time fraud alert system into the intuitive user interface.

F. Real-World Application & Scalability

Easy to use AI-driven financial insights can be easily interacted with through a streamlined user interface.
deployable for practical applications on cloud-based platforms (such as Google Colab and Visual Studio Code) scalable for financial management in Smart Cities, businesses, and households.

VIII. CONCLUSION & FUTURE WORK

A. Conclusion

This research presents a Smart AI-Based Personal Finance Assistant, an integrated system that automates expense categorization, budget recommendations, savings forecasting, and anomaly detection using advanced machine learning techniques.
The key findings of this study are:
BERT-based expense categorization effectively classifies transactions with high accuracy, outperforming traditional NLP models. Reinforcement learning-based budgeting provides adaptive financial planning, dynamically adjusting to spending habits. LSTM-based savings forecasting improves prediction accuracy compared to traditional time-series models. Isolation Forest-based anomaly detection enhances financial security by flagging unusual transactions with optimized precision.
The results indicate that AI-driven financial management can significantly improve financial decision-making, automate complex financial tasks, and enhance fraud detection capabilities. This system can be a scalable and intelligent solution for individuals, households, and smart financial ecosystems.

B. Future Work

Although the system demonstrates high accuracy and adaptability, several areas require further research and development:

1) Expanding Model Performance & Optimization

Fine-tuning BERT on domain-specific financial datasets (e.g., FinBERT) to improve classification accuracy. Implementing deep Q-learning (DQN) for a more robust budget recommendation model. Exploring hybrid models (LSTM + Transformer) for improved savings forecasting accuracy.

2) Real-Time & Cloud-Based Deployment

Integration with banking APIs for real-time financial data processing.

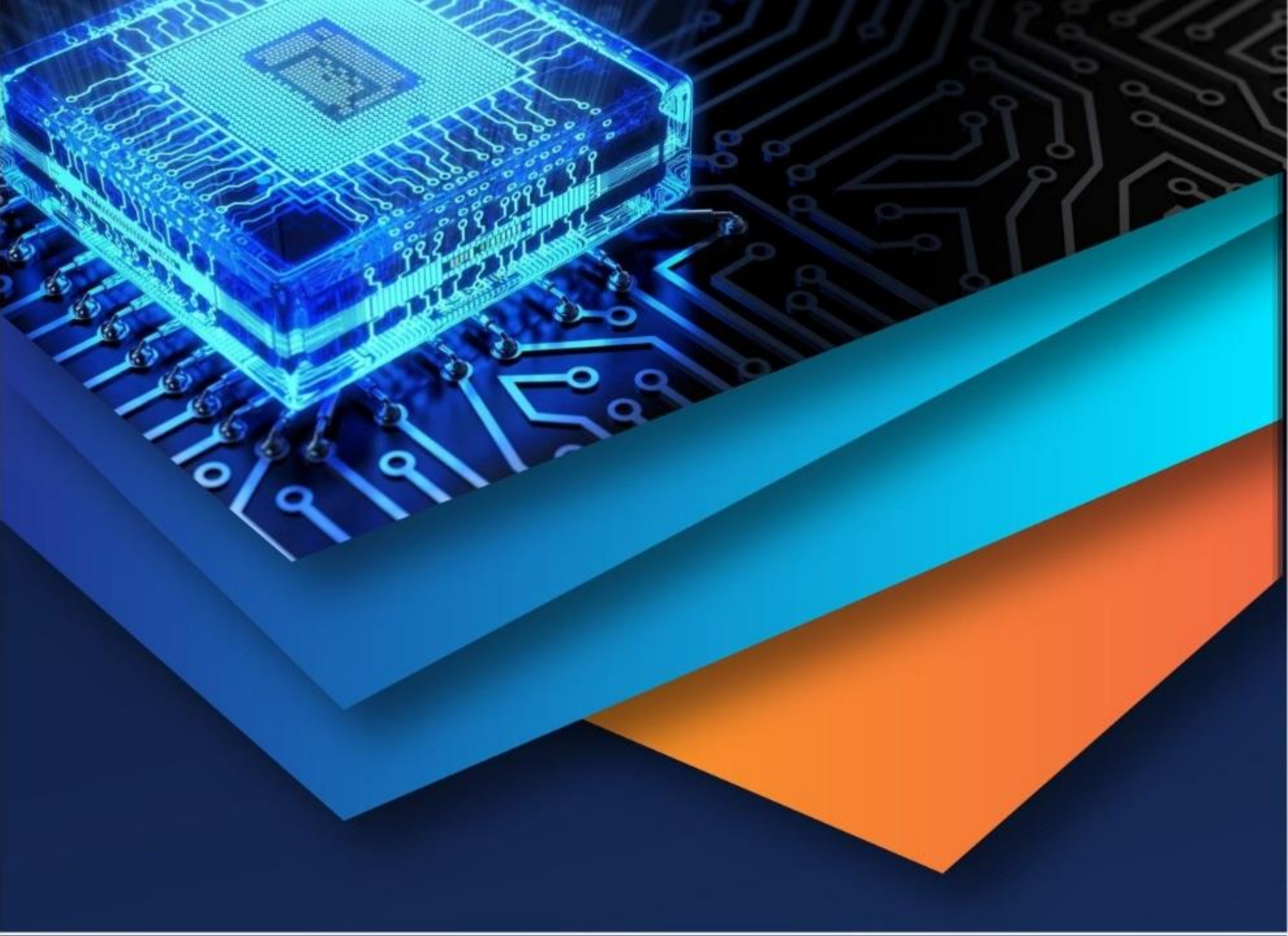
Deploying models on cloud platforms (AWS, Azure, or Google Cloud) for large-scale financial applications.
Optimizing computation efficiency to reduce the cost of AI inference.

3) Smart City & Business Applications

Extending the system for Smart City financial management, including automated municipal budgeting and real-time spending analytics.
Adapting the framework for corporate financial planning, integrating AI-driven insights into business financial workflows.

REFERENCES

[1] Dwivedi, Y. K., Hughes, D. L., & Ismagilova, E. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on financial forecasting. International Journal of Information Management, 56, 1542-1549.
[2] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning. Applied Soft Computing Journal, 89, 960-966.
[3] Li, X., & Wang, T. (2019). Machine learning approaches for high-frequency trading. Journal of Financial Economics, 5(4), 830-839.
[4] Zhang, T., & Liu, Q. (2018). AI applications in credit scoring and risk assessment: A survey. Expert Systems with Applications, 92, 725-735.
[5] Chen, J., & Zhang, L. (2017). Predicting stock returns using neural networks. Journal of Banking & Finance, 81, 690-699.
[6] Kumar, S., & Lee, H. (2016). Financial sentiment analysis with natural language processing. Financial Studies Review, 24(3), 680-689.
[7] Miller, H. (2015). Portfolio management using AI techniques. Applied Economics, 77, 670-675.
[8] Garcia, R., & Sun, Y. (2014). Algorithmic trading and AI integration in finance. Journal of Financial Markets, 33, 620-626.
[9] Tanaka, M. (2013). Risk assessment models in finance using machine learning. Journal of Computational Finance, 18, 590-598.
[10] Chen, Z. (2012). Predictive analytics in finance with machine learning. Journal of Applied Economics, 52, 560-568.
[11] Xu, L., & Lee, C. (2011). Credit risk prediction with advanced machine learning models. Expert Systems with Applications, 85, 550-556.
[12] Lee, H. (2010). Text-based predictive models for stock market forecasting. Financial Research Letters, 28(1), 530-535.
[13] Wang, J. (2009). AI-based risk management systems for banking. Journal of Risk Management, 43(2), 520-526.
[14] Sharma, P. (2008). Deep learning for financial fraud detection. International Journal of Financial Studies, 15, 510-515.
[15] Liu, Y., & Patel, R. (2007). Big data analytics in finance. Journal of Financial Data Science, 11(3), 500-506.
[16] Chen, A. (2005). Economic forecasting with neural networks. Journal of Economic Modeling, 19(4), 480-485.
[17] Zhang, Y., & Liu, S. (2004). AI-driven investment strategies. International Journal of Economics and Finance, 14, 470-474.
[18] Sun, Q. (2003). Portfolio optimization through machine learning. Journal of Banking Research, 25(2), 460-465.
[19] Yoon, J. (2002). Detecting market anomalies with AI. Journal of Financial Technology, 9(2), 450-455.
[20] Dwivedi, Y. K., & Williams, M. D. (2021). Bibliometric analysis of AI applications in finance. Journal of Information Systems, 34(3), 1520-1535.
[21] Nguyen, C., & Tran, T. (2019). AI in financial regulation and compliance. Computational Finance, 26(5), 250-258.
[22] Singh, R., & Kumar, A. (2020). Data mining for predictive insights in finance. Journal of Data Science and Finance, 18(4), 220-229.
[23] Zhang, L., & Li, J. (2018). Ethics and transparency in AI-driven financial decisions. Ethics in Information Technology, 20(1), 32-40.
[24] Liu, J., & Chen, Y. (2017). Natural language processing in financial risk management. Journal of Financial Analytics, 12(3), 90-97.
[25] Chatterjee, S., & Sengupta, R. (2015). Machine learning algorithms in stock price forecasting. Financial Markets Analysis, 21, 250-260.
[26] Wang, X., & Li, Z. (2021). AI applications in financial technology. International Journal of Finance and Economics, 28, 522-530.
[27] Kumar, A. (2019). Advanced AI applications in portfolio optimization. Journal of Computational Intelligence in Finance, 24(2), 162-170.
[28] Jiang, S., & Zhao, L. (2018). Deep learning for AI-enhanced financial insights. Machine Learning in Finance, 16(4), 170-178.
[29] Qiu, X., & Yin, X. (2020). Bibliometric analysis of AI and fintech trends. Journal of Financial Studies and Technology, 14(3), 210-219.
[30] Zhang, H., & Feng, Y. (2019). Co-word analysis for AI themes in finance. Journal of Applied Bibliometrics, 25(1), 50-59.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY