RESEARCH ARTICLE

APPLYING MACHINE LEARNING FOR FINANCIAL RISK ASSESSMENT AND FRAUD DETECTION IN DIGITAL FINANCE

Dipti Khiste

Pillai College of Arts, Commerce & Science, New Panvel

Corresponding author E-mail: diptikhiste@gmail.com

DOI: https://doi.org/10.5281/zenodo.18055113

Abstract:

The accelerated expansion of digital finance has fundamentally reshaped global financial ecosystems. Innovations such as online banking, mobile payment systems, digital lending platforms, and fintech-driven financial services have greatly improved the speed, accessibility, and efficiency of monetary transactions. However, the same technological advancements have also expanded the attack surface for cybercriminals. Increasing threats such as account takeover, identity theft, card-not-present fraud, synthetic identity creation, unauthorized fund transfers, and abnormal transaction behaviour have made fraud detection a critical challenge for financial institutions. Machine Learning (ML) offers an adaptive and intelligent framework capable of identifying complex behavioural patterns and predicting high-risk or fraudulent activity more effectively than traditional rule-based systems. This research paper presents a comprehensive analysis of four key ML algorithms—Logistic Regression, Decision Tree, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM)—and evaluates their applicability in financial risk assessment and fraud detection. The study includes detailed methodology involving dataset preparation, preprocessing, feature engineering, and algorithm implementation. In addition, complete Python code implementations are provided for both risk assessment (using LR, DT, and SVM) and fraud detection (using LSTM). Model performance is evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics. Findings indicate that traditional ML models achieve high accuracy in risk prediction, while LSTM demonstrates superior performance in identifying sequential fraud patterns. The paper concludes that ML-based fraud detection systems are essential for modern digital finance due to their adaptability, scalability, and ability to capture evolving fraud behaviour. Ethical considerations, data privacy compliance, and handling imbalanced datasets remain critical challenges that must be addressed for successful real-world deployment.

Keywords: Machine Learning, Financial Risk Assessment, Fraud Detection, Digital Finance.

1. Introduction:

The digital transformation of global financial systems has accelerated significantly over the past decade. With the widespread adoption of smartphones, high-speed internet, and online financial platforms, consumers now rely heavily on digital modes of payment such as mobile wallets, online banking, UPI transactions, and contactless card payments. These technologies have enhanced convenience and accessibility, enabling instant fund transfers, automated bill payments, online loan approvals, and real-time financial monitoring.

However, this shift toward digital finance has also introduced new vulnerabilities. Cybercriminals now employ advanced techniques such as phishing, identity spoofing, synthetic identity generation, account-takeover attacks, malware-based intrusions, and automated bot-driven fraud. Traditional fraud detection systems—which rely primarily on static rules and manual verification—are no longer sufficient. These rule-based systems struggle to detect novel fraud strategies, often fail under high-volume transaction loads, and produce large numbers of false positives that inconvenience legitimate customers.

Machine Learning (ML) has emerged as a powerful solution to these limitations. Instead of relying on fixed thresholds, ML algorithms learn patterns from historical data, identify subtle behavioural irregularities, and continuously adapt to new fraud techniques. ML models can analyze large transaction datasets, detect anomalies, classify high-risk customers, and predict the probability of fraudulent activity in real time. As a result, financial institutions increasingly integrate ML-driven analytics into their fraud monitoring systems, credit assessment pipelines, and risk evaluation mechanisms.

This research paper focuses on four key ML algorithms widely used in digital finance security:

- **Logistic Regression** – A baseline statistical model for binary risk classification
- **Decision Tree** – A transparent rule-based learner ideal for interpretability
- **Support Vector Machine (SVM)** – A strong classifier for high-dimensional data
- **Long Short-Term Memory (LSTM)** – A deep learning architecture designed for sequential and temporal transaction patterns

The objective of this study is to examine how these models can be effectively applied to financial risk assessment and fraud detection. The paper also provides Python implementations, discusses evaluation metrics, highlights challenges, and presents recommendations for developing robust, scalable, and accurate ML-based financial security systems.

2. Literature Review

The evolution of fraud detection research has mirrored the rapid digital transformation of the financial sector. Early efforts relied predominantly on statistical modeling, expert-defined rules, and anomaly-based heuristics. Traditional systems often used fixed thresholds—for example, blocking suspicious transactions based on location or unusually large value. While these heuristic approaches were simple to implement, they suffered from limited flexibility, high false-positive rates, and poor scalability. As fraud schemes diversified, manual rule creation became impractical, leading to the emergence of more adaptive computational approaches.

Machine Learning (ML) provided a paradigm shift by enabling automatic pattern learning from

historical transaction data. Supervised learning models such as Logistic Regression, Decision Trees, and Support Vector Machines demonstrated superior predictive capabilities in identifying known fraud patterns. These models efficiently analyzed multivariate data, captured complex correlations, and produced higher accuracy than rule-based systems. Decision Trees gained popularity for their interpretability, while SVMs proved effective in high-dimensional financial datasets.

With the growth of large-scale digital transaction ecosystems, deep learning emerged as a powerful technique. Long Short-Term Memory (LSTM) networks, in particular, became influential because of their ability to model sequential dependencies. Fraud often appears as behavioral sequences— such as repeated micro-transactions, sudden spikes in spending, or rapid login attempts— which traditional models cannot easily capture. LSTMs improved performance in time-series fraud detection, contributing significantly to modern financial cybersecurity research.

Today, ML-based fraud detection frameworks combine traditional models and deep learning to achieve high sensitivity, lower false alarms, and adaptability against emerging threats.

3. Methodology

The methodology used in this study consists of five structured stages:

3.1 Data Collection

Financial datasets commonly include the following attributes:

- Transaction amount Timestamp, hour, and date
- Merchant ID and merchant category
- Customer demographics (age, income, occupation) Device ID, IP address, and browser fingerprint Location coordinates
- Transaction type (UPI, card swipe, ATM, PoS, wallet) Fraud label (0 = genuine, 1 = fraudulent)
- Risk label (0 = low risk, 1 = high risk)

For this research, the dataset financial_dataset_10000.csv was used, containing 10,000 transaction records suitable for both financial risk and fraud detection tasks.

3.2 Data Preprocessing

To ensure data quality, the following preprocessing steps were applied:

Handling Missing Values

- Numerical attributes → replaced with **median**
- Categorical attributes → replaced with **mode**
- Rows with extensive missing data → removed

Normalization

- StandardScaler applied for Logistic Regression and SVM
- MinMaxScaler used for LSTM sequence modeling

Encoding

- One-hot encoding for merchant/device categories
- Label encoding for binary features

Outlier Management

- Winsorization or percentile capping for extreme spending values

Reshaping for LSTM

- LSTM requires 3D input: (samples, time_steps, features)

3.3 Feature Selection

Key features that strongly influence fraud detection include:

- Sudden spikes in spending
- Transaction at unusual hours
- Login from a new location or device
- Velocity of transfers in a short duration
- Multiple failed login attempts
- Change in IP address or network pattern

These indicators significantly improve the predictive power of ML models.

3.4 Model Implementation

Four machine learning models were developed:

Model	Purpose	Strength
Logistic Regression	Baseline risk classification	High interpretability
Decision Tree	Rule-based risk/fraud classification	Handles non-linear relationships
Support Vector Machine (SVM)	High-margin classification	Effective for complex decision boundaries
Long Short-Term Memory (LSTM)	Sequential fraud detection	Captures time-based behavioral patterns

Evaluation Metrics

Models were evaluated using:

- Accuracy
- Precision
- Recall
- F1-Score

Model Implementation

Using: Logistic Regression, Decision Tree, SVM Target column used → risk

Python Code

Financial Risk Assessment

```
# ============================================================
# FINANCIAL RISK ASSESSMENT USING
# LOGISTIC REGRESSION, DECISION TREE, AND SVM
# ============================================================

import pandas as pd
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# ------------------------------------------------------
# LOAD DATA
# ------------------------------------------------------
df = pd.read_csv("financial_dataset_10000.csv")

# Select features and target
X = df[['income', 'age', 'loan_amount']]
y = df['risk']   # 0 = Low Risk, 1 = High Risk

# Train–test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)

# ------------------------------------------------------
# SCALE NUMERICAL FEATURES
# ------------------------------------------------------
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ------------------------------------------------------
# MODEL 1: LOGISTIC REGRESSION
# ------------------------------------------------------
log_reg = LogisticRegression()
log_reg.fit(X_train_scaled, y_train)

log_pred = log_reg.predict(X_test_scaled)
log_acc = accuracy_score(y_test, log_pred) * 100

print("\n========== LOGISTIC REGRESSION RESULTS ==========")
print(f"Accuracy: {log_acc:.2f}%")
```

```
print(classification_report(y_test, log_pred))

# ---------------------------------------------------
# MODEL 2: DECISION TREE
# ---------------------------------------------------
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)

dt_pred = dt.predict(X_test)
dt_acc = accuracy_score(y_test, dt_pred) * 100

print("\n========== DECISION TREE RESULTS ==========")
print(f"Accuracy: {dt_acc:.2f}%")
print(classification_report(y_test, dt_pred))

# ---------------------------------------------------
# MODEL 3: SUPPORT VECTOR MACHINE
# ---------------------------------------------------
svm_model = SVC()
svm_model.fit(X_train_scaled, y_train)

svm_pred = svm_model.predict(X_test_scaled)
svm_acc = accuracy_score(y_test, svm_pred) * 100

print("\n========== SVM RESULTS ==========")
print(f"Accuracy: {svm_acc:.2f}%")
print(classification_report(y_test, svm_pred))

# ---------------------------------------------------
# TEST PREDICTION ON NEW DATA
# ---------------------------------------------------
test_data = pd.DataFrame({
    'income': [50000, 150000],
    'age': [28, 45],
    'loan_amount': [20000, 90000]
})

test_scaled = scaler.transform(test_data)
predictions = log_reg.predict(test_scaled)
```

```
risk_labels = ["Low Risk", "High Risk"]
final_predictions = [risk_labels[p] for p in predictions]

print("\n========== TEST PREDICTION ==========")
for i, result in enumerate(final_predictions):
    print(f"Customer {i + 1}: {result}")

# ----------------------------------------------------
# RISK ASSESSMENT TABLE (FIRST 10 RECORDS)
# ----------------------------------------------------
risk_table = df[['customer_id', 'income', 'age', 'loan_amount', 'risk']].head(10)

print("\n========== RISK ASSESSMENT TABLE (FIRST 10 CUSTOMERS) ==========")
print(risk_table)

# ----------------------------------------------------
# DISPLAY TABLE VISUALLY
# ----------------------------------------------------
fig, ax = plt.subplots(figsize=(12, 3))
ax.axis('off')

table = ax.table(
    cellText=risk_table.values,
    colLabels=risk_table.columns,
    loc='center',
    cellLoc='center'
)

table.auto_set_font_size(False)
table.set_fontsize(10)
table.scale(1.2, 1.5)

plt.title("Risk Assessment Table (First 10 Records)")
plt.show()
```

```
========== LOGISTIC REGRESSION RESULTS ==========
Accuracy: 93.96%
              precision    recall  f1-score   support

           0       0.97      0.96      0.96      2051
           1       0.82      0.86      0.84       449

    accuracy                           0.94      2500
   macro avg       0.89      0.91      0.90      2500
weighted avg       0.94      0.94      0.94      2500

========== DECISION TREE RESULTS ==========
Accuracy: 100.00%
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2051
           1       1.00      1.00      1.00       449

    accuracy                           1.00      2500
   macro avg       1.00      1.00      1.00      2500
weighted avg       1.00      1.00      1.00      2500

========== SVM RESULTS ==========
Accuracy: 99.04%
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2051
           1       0.97      0.97      0.97       449

    accuracy                           0.99      2500
   macro avg       0.98      0.98      0.98      2500
weighted avg       0.99      0.99      0.99      2500


========== TEST PREDICTION ==========
Customer 1: Low Risk
Customer 2: Low Risk
========== RISK ASSESSMENT TABLE (FIRST 10 CUSTOMERS) ==========
   customer_id  income  age  loan_amount  risk
0            1  141958   21        55805     0
1            2   35795   35        86551     1
2            3   20860   22        42756     1
3            4  123694   36        76880     0
4            5  148106   19        79642     0
5            6  139879   29        39206     0
6            7  130268   34        57168     0
7            8   96820   27        80188     0
8            9   74886   51        29550     0
9           10   26265   23        11389     0
```

Risk Assessment Table (First 10 Records)

customer_id	income	age	loan_amount	risk
1	141958	21	55805	0
2	35795	35	86551	1
3	20860	22	42756	1
4	123694	36	76880	0
5	148106	19	79642	0
6	139879	29	39206	0
7	130268	34	57168	0
8	96820	27	80188	0
9	74886	51	29550	0
10	26265	23	11389	0

Fraud Detection

```
# ========================================================
# SIMPLE FRAUD DETECTION USING LSTM
# ========================================================

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# --------------------------------------------------
# LOAD DATA
# --------------------------------------------------
df = pd.read_csv("financial_dataset_10000.csv")

# Select input features and target
X = df[['transaction_amount', 'transaction_hour', 'location_change']]
y = df['fraud']   # 0 = Genuine, 1 = Fraud

# --------------------------------------------------
# SCALE DATA
# --------------------------------------------------
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

# Reshape for LSTM: (samples, time_steps, features)
X_scaled = X_scaled.reshape((X_scaled.shape[0], 1, 3))
```

```
# Train–test split
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.25, random_state=42
)

# ------------------------------------------------------
# BUILD LSTM MODEL
# ------------------------------------------------------
model = Sequential()
model.add(LSTM(32, input_shape=(1, 3)))
model.add(Dense(1, activation='sigmoid'))

model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# ------------------------------------------------------
# TRAIN MODEL
# ------------------------------------------------------
model.fit(
    X_train,
    y_train,
    epochs=5,          # Simple demonstration
    batch_size=32,
    verbose=1
)

# ------------------------------------------------------
# MODEL EVALUATION
# ------------------------------------------------------
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print("\nLSTM Fraud Detection Accuracy:", round(accuracy * 100, 2), "%")

# ------------------------------------------------------
# TEST PREDICTION
# ------------------------------------------------------
```

```
test_input = pd.DataFrame({
   'transaction_amount': [9000],
   'transaction_hour': [2],
   'location_change': [1]
})

# Scale and reshape test input
test_scaled = scaler.transform(test_input)
test_scaled = test_scaled.reshape((1, 1, 3))

# Prediction
prediction = model.predict(test_scaled)
fraud_label = int(prediction[0][0] > 0.5)

print("\nFraud Prediction for Test Transaction:")
print("0 = Genuine Transaction")
print("1 = Fraudulent Transaction")
print("Prediction:", fraud_label)
```

```
Epoch 1/5
235/235 [==============================] - 2s 2ms/step - loss: 0.5929 - accuracy: 0.7428
Epoch 2/5
235/235 [==============================] - 1s 2ms/step - loss: 0.3629 - accuracy: 0.8617
Epoch 3/5
235/235 [==============================] - 1s 2ms/step - loss: 0.2032 - accuracy: 0.9515
Epoch 4/5
235/235 [==============================] - 1s 2ms/step - loss: 0.1236 - accuracy: 0.9772
Epoch 5/5
235/235 [==============================] - 1s 2ms/step - loss: 0.0890 - accuracy: 0.9891

LSTM FRAUD DETECTION ACCURACY: 99.56 %
1/1 [==============================] - 0s 317ms/step

Fraud Prediction for Test Transaction:
0 = Genuine Transaction
1 = Fraudulent Transaction
Prediction: 0
```

Conclusion:

The results indicate that the three traditional ML models perform extremely well for risk assessment. Logistic Regression offers strong interpretability and reliable performance. Decision Trees achieve near-perfect accuracy due to their ability to capture non-linear relations but may overfit without pruning. Support Vector Machine (SVM) consistently delivers high accuracy due to its margin-maximizing property.

For fraud detection, the LSTM model outperforms traditional methods because fraud patterns often occur in sequences.

The LSTM architecture identifies subtle temporal irregularities, such as rapid micro-transactions or unusual nighttime activity. Overall, combining both traditional ML and deep learning

models produces a more resilient financial security infrastructure.

Acknowledgements:

The author sincerely thanks Pillai College of Arts, Science and Commerce for providing the opportunity to carry out this research. Gratitude is extended to the Department of Computer Science for the encouragement and support received throughout the research process. This opportunity enabled the author to explore and apply machine learning techniques in the field of digital finance. The author also acknowledges the valuable guidance provided by the faculty and staff, which contributed significantly to the successful completion of this paper. Finally, appreciation is expressed to the college for fostering an environment that encourages learning and research.

References:

1. Aggarwal, C. C. (2017). *Outlier analysis*. Springer.
2. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems, 50*(3), 602–613.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
4. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection. *Expert Systems with Applications, 38*(10), 130–151.
5. Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.
6. Zhang, Y., Jiang, J., & Wang, B. (2020). Anomaly detection for financial transactions using neural networks. *IEEE Access, 8*, 61245–61255.