Phase-III Project Planning
Technology Stack (Architecture & Stack)

Date	06-11-2023
Team ID	Team-593025
Project Name	Project - Online Payments Fraud Detection Using ML
Maximum Marks	4 Marks

The technology stack for the Online Payment fraud detection system using ML is as follows:

a) Technical Architecture:

Data Ingestion: Data from credit/debit card transactions is ingested from various sources, such as databases or real-time streams.

Data Preprocessing: Data preprocessing involves cleaning, transforming, and normalizing the data for analysis. It may include feature engineering and dimensionality reduction.

Machine Learning Model Training: Classification algorithms like Decision Tree, Random Forest, Support Vector Machine (SVM), Extra Tree Classifier, and XGBoost are used to train the fraud detection models. This step involves model selection, hyperparameter tuning, and cross-validation.

Model Serialization: The best-performing model is serialized and saved in the ".pkl" format, which allows for easy deployment and reuse.

Flask Integration: Flask, a Python web framework, is used to create a REST API for model inference and interaction.
IBM Deployment: The model and Flask API are deployed on the IBM Cloud or an IBM Cloud service for hosting and scalability.

Real-time Scoring: The deployed model can provide real-time predictions for incoming credit/debit card transactions.

b) Open Source Frameworks:

Python: The primary programming language for data preprocessing, machine learning, and web application development.

Scikit-learn: An open-source machine learning library for training and evaluating machine learning models.

XGBoost: A popular gradient boosting framework for classification problems.

Flask: A lightweight web framework for building REST APIs.

Pandas: Used for data manipulation and analysis.

NumPy: Used for numerical computations.

Jupyter Notebook: For interactive data analysis and model prototyping.

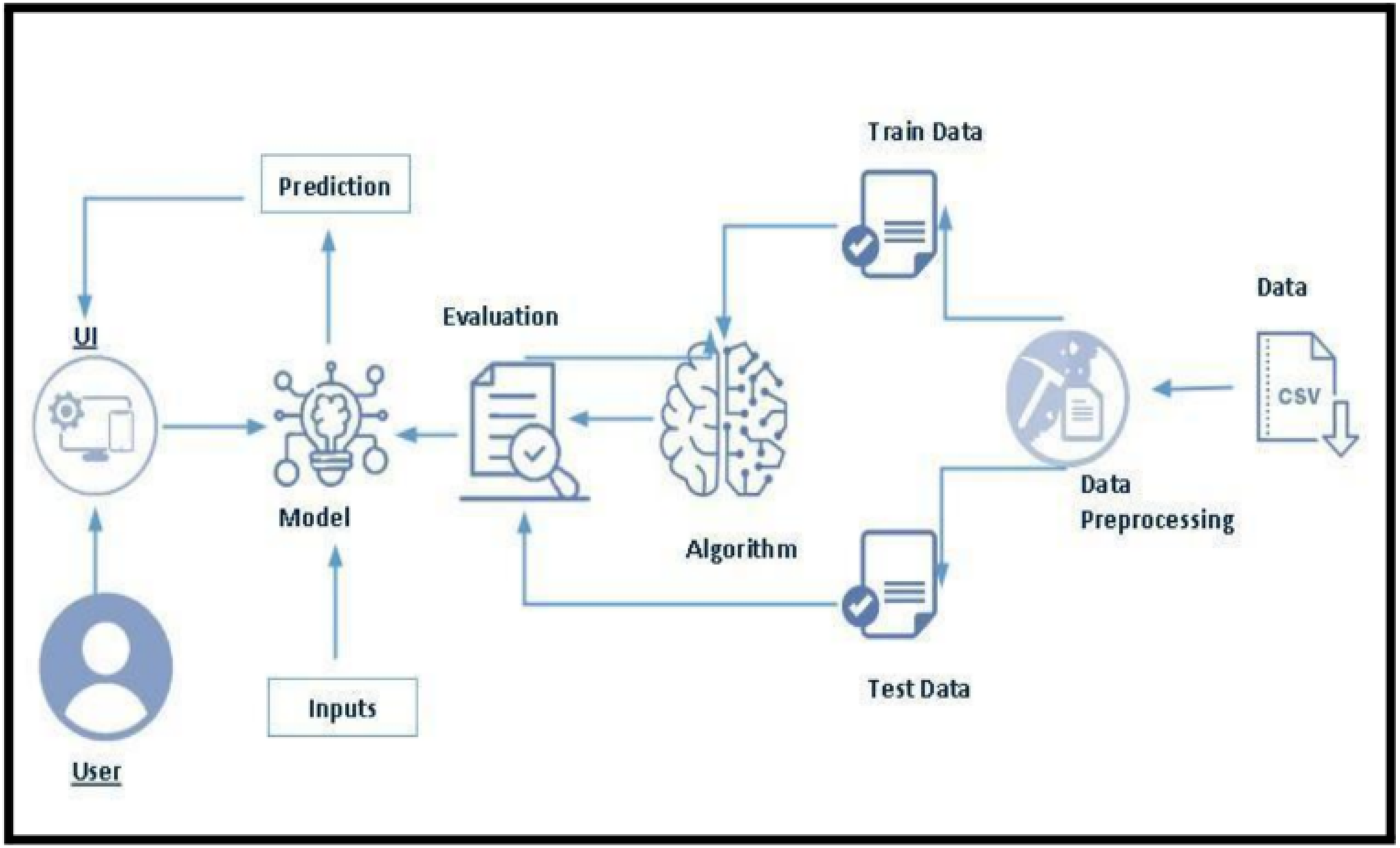
c) Cloud Deployment:

IBM Cloud: The chosen cloud platform for deploying the system, including model hosting and the Flask API. IBM Cloud provides infrastructure scalability and management capabilities.

IBM Cloud Functions: Serverless computing can be used for scaling and handling API requests efficiently.

IBM Cloud Databases: If required, databases for storing transaction history and metadata.

TECHNICAL DIAGRAM



COMPONENT	DESCRIPTION	TECHNOLOGY
Data Ingestion	Collects data from various sources for analysis.	- Python libraries (Pandas, NumPy) - Data connectors/APIs (specific to data sources)
Data Preprocessing	Cleans, transforms, and prepares data for ML.	- Python libraries (Pandas, NumPy, Scikit-learn) - Preprocessing techniques (e.g., StandardScaler)
Machine Learning	Selects and trains classification algorithms.	- Scikit-learn for ML models (Decision Trees, etc.)
Model Training		- XGBoost for boosting algorithms - Cross-validation techniques
Model Serialization	Saves the model in a reusable format.	- Python libraries (joblib, pickle)
Flask Integration	Creates a REST API for real-time predictions.	- Flask web framework
IBM Deployment	Deploys the system on IBM Cloud.	- IBM Cloud infrastructure/services

Real-time Scoring	Provides real-time predictions for transactions.	- Deployed model in memory - Web server for handling API requests

OPEN SOURCE FRAMEWORK	DESCRIPTION	TECHNOLOGIES
Python	Primary programming language.	- Python 3.x
Scikit-learn	Open-source ML library for training models.	- Scikit-learn library
XGBoost	Gradient boosting framework for classification.	- XGBoost library
Flask	Lightweight web framework for API creation.	- Flask web framework
Pandas	Data manipulation and analysis.	- Pandas library
NumPy	Numerical computations and array operations.	- NumPy library
Jupyter Notebook	Interactive data analysis and prototyping.	- Jupyter Notebook

CLOUD DEPLOYMENT	DESCRIPTION	TECHNOLOGIES
IBM Cloud	Platform for deploying applications and models. Scalability and management capabilities. Possible use of IBM Cloud Databases for storage.	- IBM Cloud infrastructure and services - IBM Cloud Functions (Serverless computing) - IBM Cloud Databases (if needed)