# **SYSTEM TESTING**

System testing is a critical phase in the software development life cycle that ensures the developed system functions correctly, efficiently, and according to the specified requirements. The purpose of testing is to detect and eliminate defects, validate system performance, and confirm that the integrated modules work harmoniously.

In the **Credit Card Fraud Detection Using Hybrid Classification Models** project, the system was tested thoroughly across multiple dimensions — including unit testing, integration testing, system testing, and validation testing — to ensure high performance, accuracy, and reliability of both the machine learning models and the web application interface.

## **OBJECTIVES OF TESTING**

1. To ensure that all system modules perform their intended operations correctly.
2. To verify that the hybrid classification model achieves the desired accuracy and robustness.
3. To ensure that the Flask web interface correctly handles user inputs and displays outputs without errors.
4. To confirm that all integration points between modules function seamlessly.
5. To validate the system’s performance under varying data sizes and input conditions.

## **TYPES OF TESTING PERFORMED**

### **1. UNIT TESTING**

Unit testing focuses on verifying the correctness of individual modules. Each component of the system — such as data preprocessing, model training, prediction, and visualization — was tested separately.

* **Modules Tested:**
  + Data Preprocessing (handling null values, scaling, and SMOTE resampling)
  + Model Training (Logistic Regression, Random Forest, SVM, XGBoost)
  + Ensemble Integration (voting/stacking mechanism)
  + Flask Routes and Input Validation

Each function was tested using **assert statements** and **mock data** to ensure the expected outputs were generated. Errors related to data format and inconsistent inputs were corrected during this stage.

### **2. INTEGRATION TESTING**

Integration testing ensures that all modules interact properly as a unified system.  
 In this project, testing focused on the integration between:

* The **machine learning model** and **Flask interface**
* The **Flask input forms** and the **prediction engine**
* The **model outputs** and **visualization functions**

Data flow between modules was verified to ensure no loss or transformation errors occurred. The hybrid model’s output was cross-checked with expected predictions during test runs.

**Result:** All integrated components functioned cohesively, and the Flask server correctly handled both batch and single prediction workflows.

### **3. SYSTEM TESTING**

System testing validates the entire system’s functionality and its compliance with project requirements. The following tests were carried out:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Type** | **Description** | **Expected Result** | **Status** |
| Functional Test | Verify each feature (upload, prediction, visualization) | All modules function correctly | ✔ Passed |
| Performance Test | Evaluate response time for batch and single inputs | Output generated within acceptable time | ✔ Passed |
| Usability Test | Check the web interface layout and clarity | User-friendly and accessible | ✔ Passed |
| Security Test | Validate file handling and input safety | No malicious input accepted | ✔ Passed |
| Compatibility Test | Run system on multiple browsers and OS | Consistent behavior | ✔ Passed |

### **4. VALIDATION TESTING**

Validation testing ensures that the developed system meets the user’s needs and expectations. The **accuracy metrics** were validated using the test dataset and cross-validation techniques.

**Validation Results:**

* **Training Accuracy:** 99.7%
* **Testing Accuracy:** 92.3%
* **Precision:** 91%
* **Recall:** 93%
* **F1-Score:** 92%
* **ROC-AUC:** 0.96

These results confirm that the system’s predictive performance aligns with research expectations and demonstrates strong generalization capability across unseen data.

### **5. USER ACCEPTANCE TESTING (UAT)**

The final stage involved evaluating the system from the user’s perspective. The test was conducted with sample datasets uploaded through the Flask interface.  
 Users verified that:

* The system correctly identified fraudulent and legitimate transactions.
* The interface displayed clear, interpretable results.
* Graphical outputs (ROC, confusion matrix) were accurate and meaningful.

**Outcome:** All stakeholders approved the system’s functionality and confirmed its readiness for deployment.

## **TESTING CONCLUSION**

The **Credit Card Fraud Detection Using Hybrid Classification Models** system successfully passed all levels of testing. The hybrid ensemble achieved strong performance metrics, and the integrated Flask web interface proved reliable and efficient. The testing phase validates that the system is accurate, robust, and ready for real-time fraud detection deployment.