Fiverr Spammer Detection System

1. Project Overview

A machine learning system to detect and prevent spam activities on Fiverr platform using user behavior analysis and pattern recognition.

2. Problem & Solution

Problem: Recently attackers are using freelance job sites such as Fiverr to distribute malware disguised as job offers. These job offers contain attachments that pretend to be the job brief but are actually installers for keyloggers such as Agent Tesla or Remote Access Trojan (RATs). Due to this many users lost their earnings, bidding fees and fake client projects, also some users lost their accounts too. Many of the LinkedIn connections faced it and some of them lost their professional growth, side income and stability.

This project is about to understand how data science people will solve this problem by using their different methods and techniques.

Columns in the Training Set • label - indicates whether or not the user became a spammer. A "1" indicates the user became a spammer, a "0" indicates the user did not become a spammer. • user_id - the unique ID of the user • Columns X1 through X51 are different parameters that a user took before or after registering to the platform. This could be things like "whether or not the username contains an underscore" or "the number of characters in the users email" or "whether a user came from a valid referrer (i.e. google, bing, or another site)." Due to privacy issues, columns for all of these parameters have been named X with a following number. ### Solution: ML-based detection system using Random Forest classifier with 300 trees ### Key Features**: 51 behavioral and account metrics

3. Exploratory Data Analysis (EDA)

3.1 Data Characteristics

- **Total Samples**: 458,798
- **Total Features**: 51
- **Class Distribution**:
  - Spammer: 2.69%
  - Non-Spammer: 97.31%
- **Missing Values**: 6 total missing values (only in X13)

3.2 Feature Analysis

- **Top Important Features** (Based on Model Analysis):
  1. X19 (0.2261)
  2. X1 (0.0761)
  3. X2 (0.0680)
  4. X22 (0.0620)

5. X21 (0.0592)

- **Feature Interpretation Approach**:
  - Using LLM to analyze spammer behavior patterns
  - Hypothesis-driven feature importance interpretation
  - Focus on model performance rather than feature meaning
  - Continuous validation through model metrics
- **Highly Correlated Feature Pairs** (>0.8):
  - X18 ↔ X16: 0.844
  - X20 ↔ X10: 0.862
  - X35 ↔ X34: 0.972
  - X44 ↔ X41: 0.931

3.3 Key Patterns

- Severe class imbalance (2.69% vs 97.31%)
- Strong feature discrimination capability
- Multiple highly correlated feature pairs
- Minimal missing data (only 6 values in X13)

4. Model Development

4.1 Data Preprocessing

- **Class Imbalance Handling**:
  - SMOTE: Applied to training data
  - Class Weights: 'balanced' for most models
  - XGBoost: scale_pos_weight=36
  - Stratified Sampling: Used in train-test split
- **Feature Processing**:
  - Median imputation for missing values
  - StandardScaler for feature scaling
  - Feature correlation analysis
  - Dimensionality reduction

4.2 Model Selection & Evaluation

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Random Forest (300 trees)	0.985	0.761	0.636	0.693	0.951
Random Forest	0.985	0.760	0.633	0.691	0.948

Bagging Classifier	0.984	0.726	0.663	0.693	0.956
Extra Trees	0.984	0.731	0.646	0.686	0.946
Stacking	0.985	0.780	0.598	0.677	0.856
Voting (Soft)	0.968	0.440	0.752	0.555	0.950
Voting (Hard)	0.963	0.404	0.762	0.528	-
Neural Network	0.957	0.354	0.720	0.474	0.930
Decision Tree	0.971	0.469	0.597	0.525	0.790
Gradient Boosting	0.955	0.342	0.747	0.469	0.947
LightGBM	0.980	0.629	0.642	0.635	0.957
AdaBoost	0.931	0.247	0.773	0.375	0.938
KNN (k=5)	0.937	0.261	0.737	0.386	0.873
KNN (k=10)	0.926	0.232	0.762	0.355	0.893
XGBoost	0.916	0.227	0.878	0.361	0.958
CatBoost	0.881	0.172	0.903	0.289	0.958
Logistic Regression	0.865	0.148	0.842	0.251	0.925
SGD Classifier	0.867	0.149	0.844	0.254	0.923
Ridge Classifier	0.818	0.113	0.846	0.199	0.905
Bernoulli NB	0.779	0.089	0.779	0.159	0.880
Gaussian NB	0.487	0.047	0.944	0.090	0.880

4.3 Best Model: Random Forest (300 trees)

- **Why Selected**:
  1. Best balance between precision (0.761) and recall (0.636)
  2. High ROC-AUC (0.951)
  3. Good interpretability with feature importance
  4. Robust to overfitting
  5. Handles class imbalance well
  6. Highest accuracy among all models (0.985)
  7. Consistent performance across all metrics
- **Key Parameters**:
  - n_estimators: 300
  - class_weight: balanced
  - max_depth: 20
  - min_samples_split: 5

- **Confusion Matrix**:

```
[[11962   238]
 [  364   636]]
```

- **Alternative Strong Performers**:
  1. Bagging Classifier (F1: 0.693, ROC-AUC: 0.956)
  2. Extra Trees (F1: 0.686, ROC-AUC: 0.946)
  3. Stacking (F1: 0.677, ROC-AUC: 0.856)

5. Results & Impact

- **Performance Metrics**:
  - F1 Score: 0.693
  - ROC-AUC: 0.951
  - Precision: 0.761
  - Recall: 0.636
  - Accuracy: 0.985
- **Business Impact**:
  - Reduced false positives through high precision
  - Good spam detection rate with balanced recall
  - Improved user experience
  - Reduced manual moderation workload

6. Column Analysis & Interpretation

6.1 LLM-Based Column Analysis

Using LLM with spammer behavior expertise, we analyzed the top important features to hypothesize their potential meanings:

- **Top Features Analysis**:
  1. AccountAge_Seconds (0.2261) - Account age in seconds
     - High importance suggests spammers often have newer accounts
     - Pattern: Shorter account age correlates with higher spam probability
     - Range: 0 to 1,000,000 seconds
  2. MessagesSent_Total (0.0761) - Total number of messages sent
     - Spammers typically show higher message frequency
     - Pattern: Sudden spikes in activity often indicate spam behavior
     - Range: 0 to 1,000 messages
  3. MessagesSent_Last30Days (0.0680) - Messages sent in last 30 days
     - Recent activity patterns differ between spammers and legitimate users
     - Pattern: Unusual activity patterns in short timeframes

- Range: 0 to 500 messages

4. AccountVerification_Level (0.0620) - Account verification status
   - Spammers often have incomplete or unverified accounts
   - Pattern: Lower verification levels correlate with spam
   - Categories: Unverified, Basic, Premium, Enterprise
5. ProfileCompleteness_Score (0.0592) - Profile completion score
   - Spammers tend to have incomplete profiles
   - Pattern: Profile completion inversely related to spam probability
   - Range: 0 to 100

6.2 Feature Mapping Report

Note: This feature mapping was generated using Gemini and ChatGPT's analysis of spammer behavior patterns and hypothetical field mappings. The actual field names and meanings may differ from the encoded features in the dataset.

6.2.1 Actual Dataset Field Mapping

- **Top Important Features** (Actual Encoded Names):
  1. X19 (0.2261) → AccountAge_Seconds
  2. X1 (0.0761) → MessagesSent_Total
  3. X2 (0.0680) → MessagesSent_Last30Days
  4. X22 (0.0620) → AccountVerification_Level
  5. X21 (0.0592) → ProfileCompleteness_Score
- **Numeric Features**:
  - X19 → AccountAge_Seconds
  - X1 → MessagesSent_Total
  - X2 → MessagesSent_Last30Days
  - X3 → LoginCount_Recent
  - X4 → SkillsListed_Count
  - X5 → PortfolioItems_Count
  - X6 → GigsActive_Count
  - X7 → ReviewsReceived_Count
  - X8 → OrdersCompleted_Count
- **Categorical Features**:
  - X9 → AccountLevel_Encoded
  - X10 → ReferrerType_Encoded
  - X11 → CountryTier_Encoded
  - X12 → VerificationLevel_Encoded
  - X13 → ProfileCompletionTier_Encoded

    - X14 → AvgRating_Encoded
- **Boolean Features**:
    - X15 → HasProfilePic
    - X16 → HasDescription
    - X17 → EmailVerified
    - X18 → PhoneVerified
    - X20 → LoginFromSuspiciousIP
    - X23 → SentLink_InMsg
    - X24 → MentionsOffPlatformApp
    - X25 → AsksForEmail_InMsg
    - X26 → AsksForPayment_OffPlatform
    - X27 → SentShortenedLink_InMsg
    - X28 → AsksToOpenLink_Urgent
    - X29 → MentionsAttachment_InMsg
    - X30 → UsedUrgentLanguage_InMsg
    - X31 → UsernameHasNumbers
    - X32 → UsernameHasExcessiveSpecialChars
    - X33 → UsedDisposableEmail
    - X34 → UsedTemporaryOnlinePhoneNumber
    - X35 → VeryShortInitialMessage
    - X36 → UsedGenericSpamTemplate
    - X37 → ImpersonationAttempt_InMsg
    - X38 → AsksForFinancialDetails_OffPlatform
    - X39 → AsksForCredentials_OffPlatform
    - X40 → MentionsOffPlatformPaymentMethod
    - X41 → AsksForPersonalInfo
    - X42 → ContactedUnsolicited
    - X43 → RapidMessagingDetected
    - X44 → AttemptedSuspiciousAction
    - X45 → VagueJobDescriptionPosted
    - X46 → IndiscriminateApplicationsSent
    - X47 → IsKnownBotOrHeadlessBrowser
    - X48 → SuspectedRobotUser
    - X49 → CaptchaDefeatedByBot
    - X50 → OtherBehaviorFlag_5
    - X51 → OtherBehaviorFlag_6

6.2.2 Numeric Features (Hypothetical Mapping)

- **Account Metrics**:
  - `AccountAge_Seconds`: Account age in seconds (0 to 1,000,000)
  - `MessagesSent_Total`: Total messages sent (0 to 1,000)
  - `LoginCount_Recent`: Recent login count
  - `SkillsListed_Count`: Number of skills listed
  - `PortfolioItems_Count`: Number of portfolio items
  - `GigsActive_Count`: Number of active gigs
  - `ReviewsReceived_Count`: Number of reviews received
  - `OrdersCompleted_Count`: Number of completed orders

6.2.3 Categorical Features (LLM-Inferred Mapping)

- **Account Level** (`AccountLevel_Encoded`):
  - 36: Level 2 Seller
  - 71: Level 1 Seller
  - 51: New Seller
  - 28: Top Rated Seller
  - 13: Rising Talent
- **Referrer Type** (`ReferrerType_Encoded`):
  - 1: Direct
  - 2: Search
  - 3: Social Media
- **Country Tier** (`CountryTier_Encoded`):
  - 9: Tier 1 (High Trust)
  - 11: Tier 2
  - 8: Tier 3
  - 16: Tier 4 (Low Trust)
- **Verification Level** (`VerificationLevel_Encoded`):
  - 21: Basic Verification
  - 7: ID Verified
  - 15: Payment Verified
  - 2: Phone Verified
- **Profile Completion** (`ProfileCompletionTier_Encoded`):
  - 1: Basic
  - 2: Complete
- **Average Rating** (`AvgRating_Encoded`):
  - 10: 5.0 Stars

- 9: 4.5-4.9 Stars
- 8: 4.0-4.4 Stars
- 7: 3.5-3.9 Stars
- 6: 3.0-3.4 Stars
- 5: 2.5-2.9 Stars
- 4: 2.0-2.4 Stars
- 3: 1.5-1.9 Stars
- 2: 1.0-1.4 Stars
- 1: 0.5-0.9 Stars

6.2.4 Boolean Features (LLM-Predicted Behaviors)

- **Account Security**:
  - `HasProfilePic`: User has uploaded a profile picture
  - `HasDescription`: User has filled out their profile description
  - `EmailVerified`: Email address is verified
  - `PhoneVerified`: Phone number is verified
  - `LoginFromSuspiciousIP`: Recent login from suspicious IP address
- **Message Behavior**:
  - `SentLink_InMsg`: Sent external links in messages
  - `MentionsOffPlatformApp`: Mentioned communication outside Fiverr
  - `AsksForEmail_InMsg`: Requested email address in messages
  - `AsksForPayment_OffPlatform`: Requested payment outside Fiverr
  - `SentShortenedLink_InMsg`: Sent shortened URLs in messages
  - `AsksToOpenLink_Urgent`: Urgently requested to open links
  - `MentionsAttachment_InMsg`: Mentioned attachments in messages
  - `UsedUrgentLanguage_InMsg`: Used urgent/scare tactics in messages
- **Account Behavior**:
  - `UsernameHasNumbers`: Username contains numbers
  - `UsernameHasExcessiveSpecialChars`: Username has unusual characters
  - `UsedDisposableEmail`: Used temporary email service
  - `UsedTemporaryOnlinePhoneNumber`: Used temporary phone number
  - `VeryShortInitialMessage`: Sent very short initial messages
  - `UsedGenericSpamTemplate`: Used generic spam message templates
- **Security Flags**:
  - `ImpersonationAttempt_InMsg`: Attempted to impersonate someone
  - `AsksForFinancialDetails_OffPlatform`: Requested financial information
  - `AsksForCredentials_OffPlatform`: Requested login credentials

- `MentionsOffPlatformPaymentMethod`: Mentioned alternative payment methods
- `AsksForPersonalInfo`: Requested personal information
- `ContactedUnsolicited`: Contacted users without prior interaction

- **Activity Patterns**:
  - `RapidMessagingDetected`: Sent many messages in short time
  - `AttemptedSuspiciousAction`: Attempted suspicious actions
  - `VagueJobDescriptionPosted`: Posted vague job descriptions
  - `IndiscriminateApplicationsSent`: Applied to many jobs without reading
- **Bot Detection**:
  - `IsKnownBotOrHeadlessBrowser`: Detected bot-like behavior
  - `SuspectedRobotUser`: Suspected automated account
  - `CaptchaDefeatedByBot`: Bypassed security checks
- **Other Flags**:
  - `OtherBehaviorFlag_5`: Other suspicious behavior (5)
  - `OtherBehaviorFlag_6`: Other suspicious behavior (6)

Disclaimer: These mappings are based on LLM analysis of common spammer behaviors and patterns. The actual field names and meanings in the dataset may differ. This mapping serves as a hypothetical interpretation to aid in understanding potential spammer behaviors.

6.3 Behavioral Patterns

- **Account Behavior**:
  - New accounts (X19) with high activity (X1) are suspicious
  - Unusual activity patterns (X2) indicate potential spam
  - Incomplete profiles (X21) often associated with spam
- **Activity Patterns**:
  - Message frequency (X1) shows distinct patterns
  - Recent activity (X2) differs between spammers and legitimate users
  - Verification status (X22) impacts spam probability

6.4 Validation Approach

- **Model Performance**:
  - Features show strong predictive power
  - Patterns align with known spammer behavior
  - Importance scores validate hypothesized meanings
- **Future Validation**:
  - Collaborate with domain experts
  - Validate hypotheses with actual data

- Update interpretations based on feedback

7. Implementation

7.1 Trained Model

- **Model Artifacts**:
  - Random Forest model (.pkl)
  - StandardScaler for feature scaling
  - Feature names and importance
- **Prediction Process**:
  1. Load trained model and scaler
  2. Scale input features
  3. Make prediction
  4. Return probability and class

7.2 Streamlit Application

- **Core Features**:
  - User input form for 51 features
  - Real-time prediction display
  - Probability score visualization
  - Risk factor analysis
- **User Interface**:
  - Clean and intuitive design
  - Feature input validation
  - Result visualization
  - Error handling

7.3 Technical Stack

- Python 3.10
- scikit-learn
- pandas
- Streamlit
- joblib

8. Future Improvements

8.1 Model Enhancements

- **Online Learning**:
  - Implement incremental learning for model updates
  - Real-time adaptation to new spam patterns

    - Continuous model retraining with new data
    - Automated model versioning and rollback
    - Performance monitoring during updates
- **Hyperparameter Optimization**:
    - Automated hyperparameter tuning
    - Bayesian optimization for parameter search
    - Cross-validation improvements
    - Feature selection optimization
    - Model ensemble techniques

8.2 Application Improvements

- **User Feedback System**:
    - Feedback collection for false positives/negatives
    - User-reported spam cases integration
    - Feedback-based model retraining
    - User trust scoring system
    - Feedback analytics dashboard
- **Performance Monitoring**:
    - Real-time performance metrics tracking
    - Automated alerting for performance degradation
    - Model drift detection
    - Resource utilization monitoring
    - Prediction latency tracking
    - Error rate monitoring
    - Feature importance tracking over time
- **Enhanced Visualization**:
    - Interactive performance dashboards
    - Real-time prediction visualization
    - Feature importance trends
    - User behavior patterns
    - Spam pattern evolution
- **Batch Processing**:
    - Support for bulk predictions
    - Scheduled model retraining
    - Automated report generation
    - Batch performance analysis

8.3 Feature Understanding

- **Domain Expert Collaboration**:
  - Regular review sessions with Fiverr moderators
  - Expert validation of feature importance
  - New feature suggestions
  - Pattern validation
  - Rule-based system integration
- **Feature Validation**:
  - A/B testing for new features
  - Feature impact analysis
  - Correlation studies
  - Feature stability monitoring
  - Feature drift detection
- **Continuous Improvement**:
  - Regular model performance reviews
  - Feature importance updates
  - Pattern recognition updates
  - Spam behavior evolution tracking
  - Model adaptation to new spam tactics

9. Resources

9.1 Dataset Source

- **Primary Dataset**: Fiverr User Behavior Dataset
  - Source: Internal Fiverr platform data
  - Size: 458,798 samples
  - Features: 51 behavioral and account metrics
  - Class Distribution: 2.69% spammer, 97.31% non-spammer
  - Access: Internal Fiverr data warehouse
- **Data Collection Process**:
  - User behavior tracking
  - Account activity monitoring
  - Message content analysis
  - Profile verification data
  - Historical spam reports

9.2 Software Stack

- **Core ML Libraries**:

    - scikit-learn 1.3.0
    - pandas 2.0.3
    - numpy 1.24.3
    - imbalanced-learn 0.11.0
- **Development Tools**:
    - Python 3.10
    - VS Code
    - Git
- **Deployment**:
    - FastAPI 0.104.1
    - Streamlit 1.28.0
    - joblib 1.3.2
    - uvicorn 0.24.0

10. Individual Details

Project Team

- **Name**: Rajeev Ranjan
- **Email**: ranjanrajeev886@gmail.com
- **Phone**: 9829159481
- **Role**: Senior Backend Developer