# Machine Learning-based Financial Fraud Text Message Detection System

Name:Lu Pengran, Chen Chi, Wang Haoyu,Wang Yican
Supervisor:Liu Qi

## 1 Introduction

With the widespread use of smartphones, more and more things can be easily done on the mobile phone, like taking a ride or paying bills. However, this convenience has also brought many issues, such as the exposure of privacy, which can provide opportunities for illegal activities to take advantage. The most concerning aspect is the recent proliferation of SMS scams. Scammers often impersonate various departments or organizations, claiming that there is an issue with your account and asking you to click on a specific link to resolve it. As these links are fraudulent, clicking on them can lead to financial theft, as scammers exploit these opportunities to illicitly obtain funds. For example, a recent scam involved fraudsters impersonating HKeToll, where users were prompted to click a link to resolve an alleged payment irregularity. This ultimately resulted in a substantial unauthorized transaction from the user's account[1].

Currently, there are numerous software solutions available for detecting these scam messages, which typically rely on manually flagged fraudulent phone numbers to filter and block such scams. However, with the rise of virtual phone numbers, scammers can leverage specific applications to generate thousands of phone numbers at no cost daily. With this endless supply of virtual numbers, scammers can easily bypass detection software by slightly altering their messaging contents. As a result, these tools have proven to be less effective in protecting users.

To address this issue, we design a Machine Learning-based Financial Fraud Text Message Detection System. This system applies various algorithms to detect and block spam and fraudulent messages. Even when scammers frequently update virtual phone numbers, the system can accurately capture key features to intercept them, which significantly improves the accuracy of detection while reducing the cost of manual review.

# 2 Aims and Objectives

## 2.1 Aims

This project aims to address the problem of text message fraud through a Machine Learning-based Financial Fraud Text Message Detection System.

## 2.2 Objectives

The objectives of this project are:

1. to provide an automatic detection system that combat the rapidly evolving fraud techniques.

2. To reduce the need for manual methods like flagging or reporting scam messages.

3. To protect users from large financial losses by stopping fraudulent transactions caused by text message scams.

# 3 Literature Review

## 3.1 Current Situations

Fraud text detection has become a critical area of focus with the rise of digital communications and online transactions. Traditional approaches to detecting fraud, such as rule-based systems and classical machine learning models, often struggle to keep up with the evolving nature of fraud patterns. These systems rely heavily on predefined rules or features, which can be easily circumvented by more sophisticated attacks.

Machine learning based algorithms were very popular 4 years ago. By statistical measures, the pattern of spam and non-spam messages can be learned and adopted for a better decision. According to the study[2] of Dr. R. Al-Haddad and her colleagues, algorithm-based models such as SVM and random forest can already perform very well in terms of spam detection, with SVM being the best and reaching an accuracy of 98% for a sample of 11,926 emails in which 5,183 are spam.

Recent advancements in generative AI and large language models (LLMs), such as GPT has made such detection difficult to proceed, as they can imitate the words from humans. However, with proper training, they can also be very helpful, and provide huge support in terms of the detection of spam mails and text messages.

## 3.2 Gaps from just LLMs

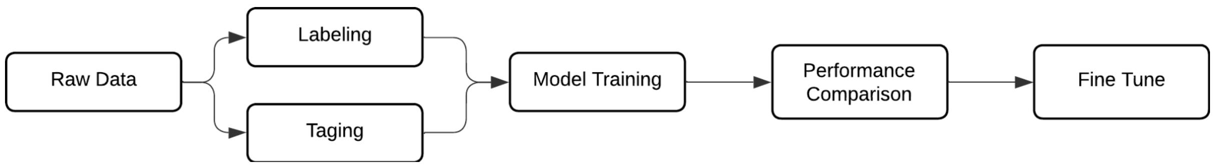
Through approaches such as reverse engineering, it is possible to detect whether a text is generated by AI or not. It is also feasible to leverage those models to detect fraud and AI generated messages. Amrita Bhattacharjee and Huan Liu’s findings[3] provide insight on how ChatGPT and similar LLMs may be leveraged in automated detection pipelines by simply focusing on solving a specific aspect of the problem and deriving the rest from that solution.

One very significant problem for this is that they all, to a certain extent, rely on cloud services to function. The elderly are those that will be affected most since they possess the least internet exposure and are generally those that suffer the most from fraudulency. Also, the fact that some models (e.g., ChatGPT) do not provide services to some regions and countries makes it hard to implement the measure globally.

## 3.3 Justification for our Project

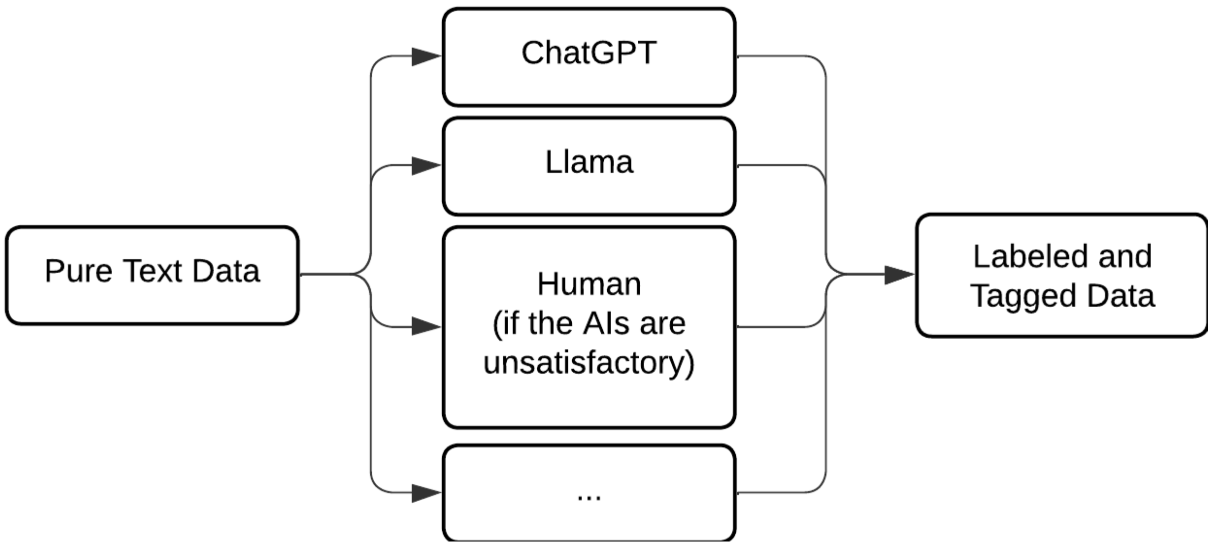
As we plan to move the model locally, we are bound to lose the flexibility and possibility that LLMs can bring. As a result, we will exploit them in terms of labeling and tagging to make sure the algorithms can receive the best input data for training. By combining both, the efficiency and feasibility of the model can be improved.

## 4 Methodology



Here is the pipeline to proceed with the project. First we will get the raw data from web scraping and some existing datasets. Then we label them and assign them with reasonable tags. Then they will be passed to a few models for training. The performance of each model will be noted for comparison, and we will fine tune the best one to make it effective but still able to run locally.

## 4.1 Data Processing



The study[4] conducted by Yiming Zhu and his colleagues has demonstrated that ChatGPT has some ability on labeling data. They compared the work done by ChatGPT and benchmarked it with the human labeling, and found out that ChatGPT obtains an average accuracy of 60.9%, with a highest accuracy of 64.9% in sentiment analysis. As a result, the approach would be that we use several AI for labeling to minimize errors.

The primary assumption here, as a result, will be that the models should give accurate results. If the result looks unsatisfactory, we will also consider labeling them by ourselves.

It is never true that the text is either fraud or normal. We will also need to look at the intermediate levels such as promotional text. As a result, we will add tags based on the content of the text so that we can better analyze and classify them.

Also, we will need to make some modifications on the text itself. For example, the removal of commas, full stops, and other punctuation marks is important for model training as they will, to some extent, confuse the model. We will dig deeper on the selection of words that will also be removed. These include some particles, any words that are not related to the content of the text, etc. Again, LLMs can be utilized to simplify the process.

## 4.2 Model Training

We actually are not very sure on which model to choose in terms of training, so we plan to spend time trying out a variety of models and compare the results.

1. Support Vector Machine

   Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification and regression tasks. The key idea behind SVM is to maximize the margin between the closest data points (called support vectors) from each class and the hyperplane, ensuring a robust separation.

   SVM easily outperforms other machine learning algorithms in high-dimensional spaces in terms of accuracy. SVMs are particularly effective because they use a hyperplane to separate data points into different classes, maximizing the margin between them. This approach reduces the risk of overfitting and improves generalization performance. [5] The time for training may be affected though.

2. Random Forest

   A decision tree is an algorithm that learns to produce branches when an entry of training data is inputted. It will give predictions according to the branches of rules it learnt from training data. A random forest consists of multiple decision trees, and the result of classification is produced where each tree votes and counts.

   However, when the dimensionality of data increases, Random Forest performance can decline. This is due to the curse of dimensionality, where the sparsity of data in high-dimensional spaces reduces the effectiveness of splitting criteria within decision trees. We will dig further in the model training part.

3. Naive Bayes

   Naive Bayes is a probabilistic classifier that uses Bayes' Theorem and assumes independence between features. The formula is:

$$P(C|X) = \frac{P(X|C)\cdot P(C)}{P(X)}$$

   where $P(C|X)$ is the posterior probability of class $C$ given features $X$, $P(X|C)$ is the likelihood, $P(C)$ is the prior probability of class $C$, and $P(X)$ is the evidence.

   Naive Bayes works well in low-dimensional spaces but may underperform as dimensionality increases if features are highly correlated, violating the independence assumption. Despite this, it remains efficient, scaling well with large datasets, and is commonly used in high-dimensional tasks like text classification.

4. Logistic Regression

   Logistic Regression is a linear model used for binary classification that predicts the probability of a class label using the logistic function (sigmoid). The model estimates the probability $P(y = 1|X)$ as:

$$P(y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\cdots+\beta_n X_n)}}$$

   Logistic Regression performs well for small to moderate dimensions but can struggle in high-dimensional spaces due to overfitting and collinearity among features. Regularization techniques like L1 (Lasso) and L2 (Ridge) are often applied to improve performance by penalizing large coefficients, making them more robust in high dimensions.

5. LSTM

   Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to model sequential data by capturing long-term dependencies. Unlike standard RNNs, LSTM uses memory cells and three gates—input, forget, and output—to regulate information flow, allowing it to retain or discard information as needed. The key formulas for LSTM are:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

LSTMs perform well in high-dimensional data, especially for time-series or text, by mitigating the vanishing gradient problem. However, high dimensionality can lead to overfitting or increased computational costs, making regularization and optimization crucial for large datasets.

6. BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) extends the LSTM architecture by processing the input data in both forward and backward directions, capturing context from both past and future sequences. This is especially useful in tasks like speech recognition, language modeling, and sentiment analysis, where the meaning of data points may depend on both preceding and succeeding elements. The architecture consists of two LSTM layers: one processes the sequence from the start to the end, while the other processes it from the end to the start. The outputs from both directions are then combined. The formulas for the forward and backward LSTMs are:

$$h_t Forward = LSTM_f(x_t, h_{t-1} Forward),\ h_t Backward = LSTM_b(x_t, h_{t-1} Backward)$$

BiLSTM performs effectively in high-dimensional data, especially in NLP, as it captures richer context. However, like LSTMs, it faces challenges such as overfitting and increased computational costs with growing dimensions, necessitating regularization strategies.

While these two NLP algorithms may perform well, we will also monitor their computational cost and evaluate the feasibility for them to run locally on most devices.

7. BERT and ChatGPT

After trying out these models, we will also test the performance of BERT and ChatGPT and compare the results.

## 4.3 Model Performance

The performance of models will be assessed mainly by 5 criteria.

1. Precision-weighted Average

   Precision is defined as the ratio of true positive predictions to the total positive predictions (true positives + false positives). It indicates how many of the predicted positive instances were actually positive.

$$PrecisionWeighted = \frac{\frac{TP}{TP+FP} \times P + \frac{TN}{TN+FN} \times N}{P+N}$$

   The higher the value, the more precise the model, particularly on those most frequent classes.

2. Recall-weighted Average

   Recall measures the ability of a model to identify all relevant instances (true positives) from the actual positives. It is particularly important in contexts where missing positive instances is costly.

$$RecallWeighted = \frac{\frac{TP}{TP+FN} \times P + \frac{TN}{TN+FP} \times N}{P+N}$$

   The higher the value, the more effective the model is at capturing positive instances across the dataset.

3. F1-score Weighted Average

   F1-score is the Harmonic mean of precision and recall. The F1 Score measures how well our model balances between correctly identifying spam text (precision) and catching all the spam text that were actually there (recall).

$$F1ScoreWeighted = \frac{\frac{2 \times Precision_P \times recall_P}{precision_P + recall_P} + \frac{2 \times Precision_N \times recall_N}{precision_N + recall_N}}{P+N}$$

The higher the value, the stronger the model is at maintaining the balance between precision and call, which further implies that it can effectively manage both false positives and false negatives.

4. False Negative Rate

   False negative rate is the ratio of false negatives to the total actual positives. It measures the proportion of positive instances that were incorrectly predicted as negative.

$$FNR = \frac{FN}{TP+FN}$$

   The lower the FNR, the more effective the model is at identifying positive instances.

5. True Negative Rate

   True negative rate is the ratio of true negatives to the total actual negatives. It measures how well the model identifies negative instances.

$$TNR = \frac{TN}{TN+FP}$$

   The higher the TNR, the better the model is at correctly identifying negative instances, reducing false alarms.

By comprehensively considering all the criteria, we will select the best model for prediction. We will stick to the model for find-tuning and delivering.

We will also show the result of directly using the LLMs to identify if a text message is fraud or not to compare the results.

## 4.4 Fine Tuning and Delivering

We will spend the rest of the time fine tuning the selected model as well as testing. We will also produce an interactive application to let users input any message.

# 5 Products and Deliverables

---

## 5.1 Data Collection and Preprocessing Module

The data collection and preprocessing module will extract and clean data from user-input SMS and SMS packets obtained by web crawlers. The module will perform basic text processing and feature extraction. The processed data will be used for the model's real-time detection.

## 5.2 Fraudulent SMS Detection Program

The Fraud SMS Detection Program is the core component of this project. This component is responsible for processing user-input SMS in the local environment. Users can run the program directly on their local devices without relying on external servers. It will use some machine learning models to analyze the SMS content and return the detection results to the user.

## 5.3 Machine Learning Model

The machine learning model is the core of this platform and is responsible for classifying the SMS based on the pre-trained dataset. The model will implement multiple algorithms that the user can train and test locally. The model will be embedded into the detection program to process the user's SMS requests in real-time.

## 5.4 Result Display Interface

The result display interface will display the detection results in real time so that users can quickly check whether an SMS is labeled as a scam after entering it. The interface will support visualization of the possibility of fraudulent SMS, so the user can easily understand the detection results.

## 5.5 User Feedback System

The user feedback system will allow users to subjectively evaluate the detection results in order to optimize the machine learning model continuously. Users can submit feedback, and this information will be used for subsequent model improvements and adjustments to improve detection accuracy.

# 6 Scope and Timeline

---

## 6.1 Scope

The scope of this project includes:

- Implementing a data collection and preprocessing module responsible for extracting and cleansing user-input SMS data.
- Development of a locally run fraudulent SMS detection program.
- Develop a machine learning model to support multiple algorithms for SMS detection.
- Creating a result display interface to show the real-time SMS detection results.
- Create a user feedback system to collect users' evaluations and feedback on the detection results.

## 6.2 Timeline

1. Requirements analysis and design

- Time: Week 1 - Week 2
- Tasks: Determine project requirements and design system architecture.

2. Data collection and preprocessing module development

- Time: Week 3 - Week 4
- Tasks: Realize data extraction, cleaning, and feature extraction functions.

3. Machine learning model development

- Time: Week 5 - Week 7
- Tasks: Select and implement various machine learning algorithms, model training, and testing.

4. Fraudulent SMS detection program development

- Time: Week 8 - Week 9
- Tasks: Integrate machine learning models and develop the core functions of the SMS detection program.

5. Results display interface development

- Time: Week 10 - Week 11
- Tasks: Design and implement the user interface to ensure the results are displayed promptly.

6. User feedback system development

- Time: Week 12
- Tasks: Analyze the feedback collection function and design the feedback processing process.

7. System testing and optimization
- Time: Week 13 - Week 14
- Tasks: Conduct comprehensive testing and optimize program performance and user experience.

8. Project Documentation and Report Writing
- Time: Week 15 - Week 16
- Tasks: Write the project summary report and organize related documents.

# 7 Project Management

---

## 7.1 Roles and Responsibilities

•**Lu Pengran, Chen Chi**: They will research and construct machine learning models for detecting financial fraud-related text messages.Including selecting suitable machine learning algorithms, model training and optimization, performance evaluation, and iterative adjustments based on requirements in order to ensure the development of an efficient and accurate text detection system for identifying financial fraud messages effectively.

• **Wang Haoyu, Wang Yican**: They will be responsible for gathering datasets relevant to financial fraud and other kind of spam messages, annotating the data, and preprocessing it to provide usable datasets for the modeling team which including collecting text data from various sources, cleaning, annotating, feature extraction, and transformation to facilitate model training and evaluation in order to provide high-quality, accurately labeled datasets to support the modeling team's needs in building the fraud detection model.

• **Testing:**After the data and model completion, we will run the test process to validate the efficacy and accuracy of the developed project. The testing process will involve the systematic evaluation of the model's performance in identifying financial fraud-related text messages. This step includes creating comprehensive test cases, executing tests, analyzing results, and providing feedback for model refinement. Each team member will participate in the testing phase to ensure a thorough assessment of the system's functionality and to address any potential issues or areas for improvement. The testing phase aims to validate the system's effectiveness in detecting fraudulent text messages accurately, ultimately contributing to the overall success of the project.

• **Other Responsibilities:**Each team member will participate in regular team meetings, contribute to problem-solving discussions, assist in testing and debugging the system, and collaborate on the preparation and presentation of project reports and documentation.

## 7.2 Communication Plans

• **Regular Meetings:**There will be weekly meetings holding to share the project’s process and addressing any issues or challenges relating to the project.
• **Instant Messaging:**There will be a platform for us to share up to date information about the project in a timely manner.This will provide a stable way to quick message exchange and problem-solving discussions.
• **Collaboration Tools:** We will use Google doc for documents collaboration.We will use Github and the platform within the FYP moodle page to help construct our code.These platforms allow us to work together simultaneously and track changes.
• **Individual Responsibilities:** Every team member is accountable for effectively communicating their advancements, obstacles, and requirements.Team members must communicate promptly if they have difficulty meeting a deadline or completing a task.
• **Conflict Resolution:**If disagreements or conflicts arise, we will strive to resolve them through transparent discussions and compromises.As a resolution method, we may use a majority vote if necessary.

# 8 Expected Challenges and Solutions

---

## 8.1 Data Quality and Acquisition

**Challenges:** Collecting high-quality SMS datasets can be difficult, especially samples that are labeled as fraudulent and legitimate.

**Solutions:** Consider using public datasets and combining them with manual annotations to increase the diversity and accuracy of the dataset.

## 8.2 Selection and Tuning of Machine Learning Models

**Challenges:** Selecting the right machine learning algorithm and performing effective model tuning requires time and experience.

**Solutions:**Test multiple algorithms and optimize the model using cross-validation and hyperparameter tuning.

## 8.3 Effectiveness of feature extraction

**Challenge:** Effective feature extraction is critical to model performance, and features of text data can be difficult to extract.

**Solution:** The best solution would be to investigate and use a variety of text processing techniques, like TF-IDF, word embedding, etc., to determine which will provide the best features.

## 8.4 User experience design

**Challenge:** Ensuring that the user interface is easy to use and can effectively display detection results can be complex.

**Solution:** Testing and collecting user feedback are the best ways to improve the user interface. The design should be iterated to improve the user experience.

## 8.5 Real-time performance

**Challenge:** When running locally, ensuring that the program can process text messages and return results in real time may be limited by computing resources.

**Solution:** Implement code and algorithms that are optimized for resource-limited environments to ensure efficient performance.

## 8.6 Misjudgment and user feedback

**Challenge:** The model may make misjudgments, causing users to distrust the results.

**Solution:** Providing transparent detection evidence and setting up an effective feedback mechanism for users is the solution.

## 8.7 Project time management

**Challenge:** Completing all tasks within the specified time may be stressful, especially when encountering technical difficulties.

**Solution:** Develop a detailed project plan, check progress regularly, and make adjustments when necessary to ensure timely completion.

## 9 References

[1] I.A. Glover and P.M. Grant, *Digital Communications*, 3rd ed. Harlow: Prentice Hall. 2009, https://www.thestandard.com.hk/section-news/section/4/256314/Warning-on-fraudulent-HKeToll-SMS-messages

[2] R. Al-Haddad, F. Sahwan, A. Aboalmakarem, G. Latif and Y. M. Alufaisan, "Email text analysis for fraud detection through machine learning techniques," 3rd Smart Cities Symposium (SCS 2020), Online Conference, 2020, pp. 613-616, https://doi.org/10.1049/icp.2021.0909.

[3] Amrita Bhattacharjee and Huan Liu. 2024. Fighting Fire with Fire: Can ChatGPT Detect AI-generated Text? SIGKDD Explor. Newsl. 25, 2 (December 2023), 14–21. https://doi.org/10.1145/3655103.3655106

[4] Zhu, Y., Zhang, P., Haq, E.-U., Hui, P., & Tyson, G. (2023). Can ChatGPT reproduce human-generated labels? A study of social computing tasks. arXiv. https://doi.org/10.48550/arXiv.2304.10145

[5] The power of support vector machines: Effectiveness in high dimensional spaces. redShift Recruiting. (n.d.). https://www.redshiftrecruiting.com/support-vector-machines-high-dimensional-spaces#:~:text=In%20high%2Ddimensional%20spaces%2C%20support,maximizing%20the%20margin%20between%20them