THAISMS SCAMGUARD

DADS 7203: Text Analytics and Natural Language Processing

OBJECTIVE

To develop a deep learning-based system, ThaiSMS-ScamGuard, for accurately classifying Thai SMS messages as “Scam“ or “Non-Scam“ using a fine-tuned OpenThaiGPT model.

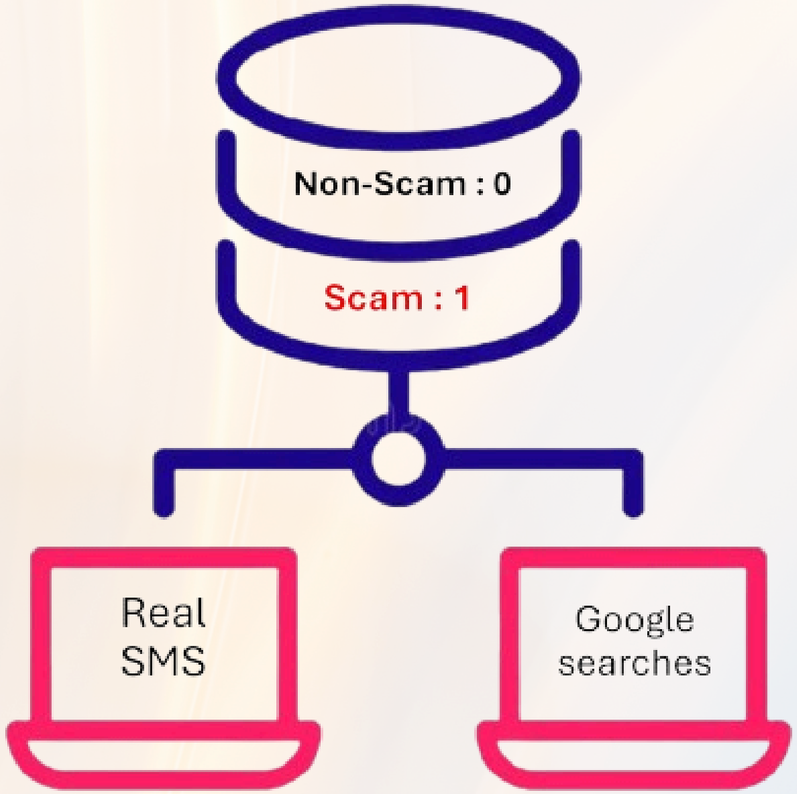
DATASET

The dataset for ThaiSMS-ScamGuard was compiled from 2 main sources

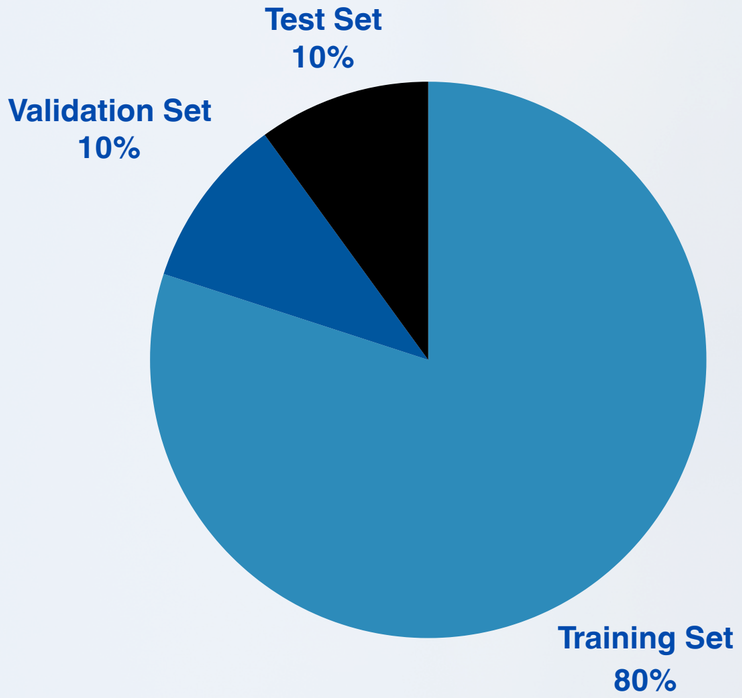
- Team Contributions: Real SMS messages received by team members.
- Public Sources: Datasets found via Google searches.

All messages were labeled as either “Scam” or “Non-Scam.”
The data distribution is outlined below

Class	Count
0: Non-Scam	309
1 : Scam	306
Total	615



DATA SPLITTING

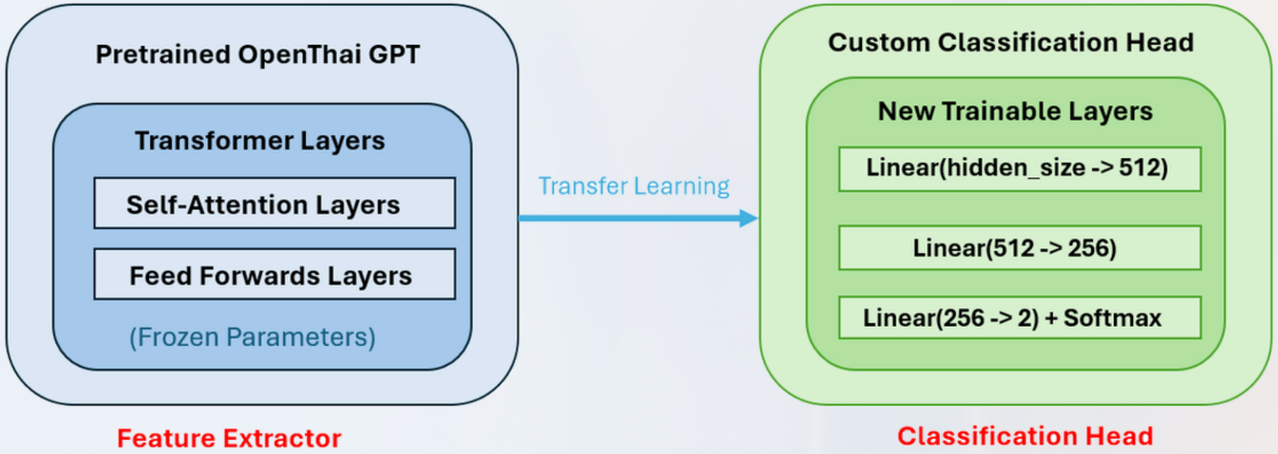


The dataset was divided using a stratified train-test-validation split to maintain class balance across subsets, essential for binary classification. The splits are:

- Training Set: 70%
- Validation Set: 10%
- Test Set: 20%

Stratification (based on labels) ensured consistent “Scam“ and “Non-Scam“ distribution across all splits.

MODEL ARCHITECTURE



This research utilized the pre-trained OpenThaiGPT 1.5 7B as a feature extractor due to its superior performance among Thai language models. OpenThaiGPT 1.5, based on Qwen v2.5 and fine-tuned on over 2 million Thai instructional pairs, demonstrates state-of-the-art results across diverse Thai language tasks.

A transfer learning approach was adopted for SMS scam detection:

- Feature Extractor: The entire OpenThaiGPT 1.5 model was used with its weights frozen.
- Classifier Head: A custom fully connected layer was added and trained for binary classification (“Scam“ vs. “Non-Scam“).

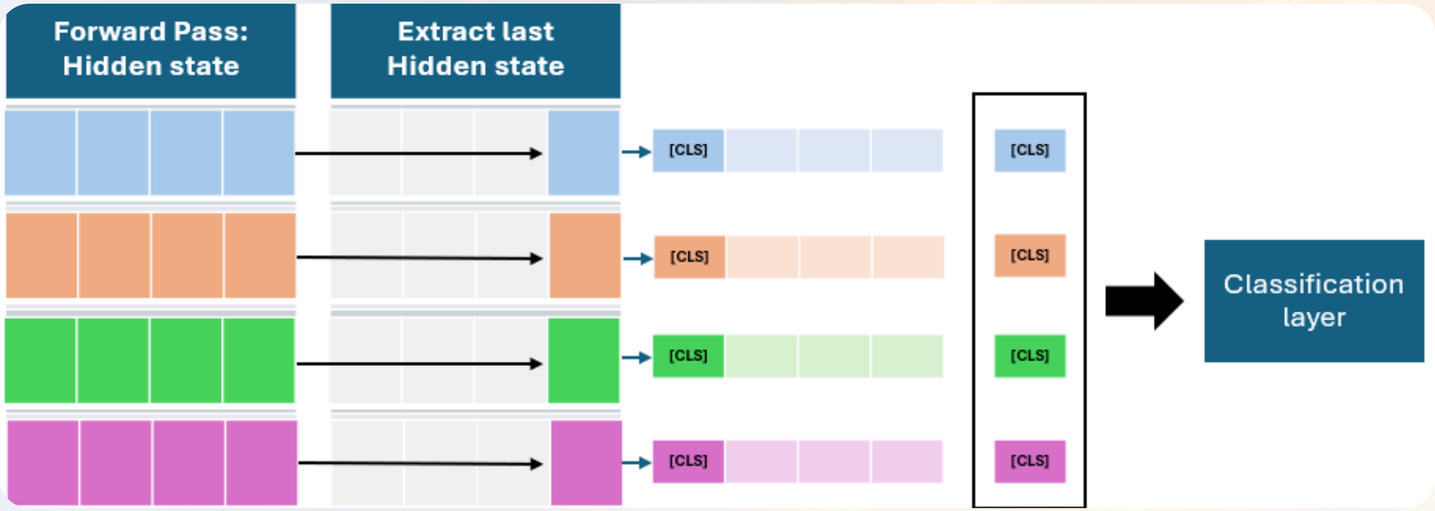
This strategy leveraged pre-trained knowledge while adapting the model for the specific task of SMS scam detection.

FEATURE EXTRACTOR

Feature Extraction Process for Presentation:
Key feature extraction in ThaiSMS-ScamGuard occurs during the forward method:

1. **Forward Pass:** Hidden states from all layers of OpenThaiGPT are obtained by setting output_hidden_states=True.
2. **Extract Last Hidden State:** The last hidden state, which contains the most processed information, is extracted.
3. **[CLS] Token Representation:** The representation of the [CLS] token, summarizing the entire sequence, is taken as the final feature vector.
4. **Classification Layers:** The [CLS] vector is passed through the classifier layers to predict the SMS label.

This structured approach ensures efficient feature aggregation for accurate SMS classification.

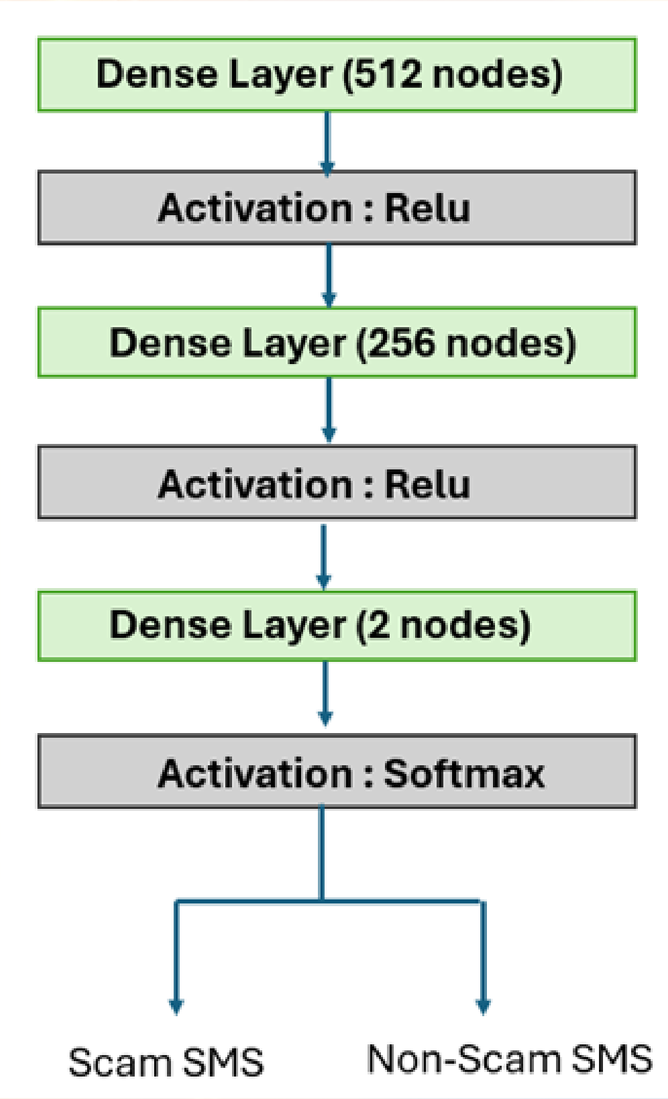


CLASSIFICATION HEAD

The custom classification head transforms the [CLS] token representation into class probabilities:

1. Layer Structure:
   - Two fully connected layers with 512 and 256 nodes respectively.
   - A final layer with 2 nodes using a Softmax function to predict “Scam“ and “Non-Scam“ classes.
2. Activation Function:
   - ReLU (Rectified Linear Unit) applied in the fully connected layers for computational efficiency and non-linearity.

This head processes the [CLS] token to produce accurate class probabilities for binary classification. (See Figure 3 for illustration.)



TOKENIZATION

The OpenThaiGPT pre-trained tokenizer prepares Thai text for the model by:

- Converting text to token IDs for neural network processing.
- Truncating longer sequences to 512 tokens to ensure compatibility with the model.
- Padding shorter sequences to 512 tokens using padding tokens, ensuring uniform input lengths.

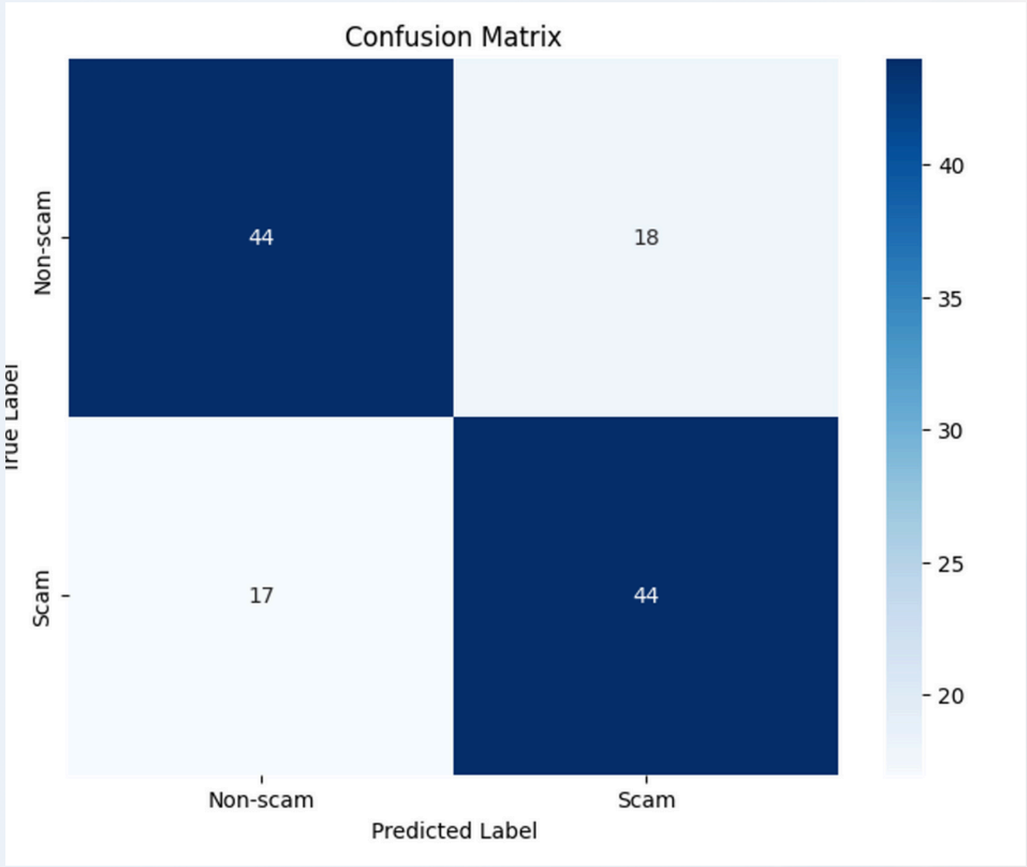
This process handles preprocessing efficiently, including creating attention masks for accurate model input.

MODEL TRAINING

1. **Training Setup**
   - Loss Function: Binary cross-entropy for error calculation.
   - Optimizer: Adam, with a learning rate of 0.001.
   - Training Setup: 10 epochs, batch size = 16.

2. **Early Stopping Strategy**
   - Prevents overfitting by monitoring validation loss.
   - Training stops if validation loss doesn't improve for 3 consecutive epochs (patience).
   - Model stopped training after 5 epochs due to early stopping.
   - This approach ensures efficient training while maintaining generalization to unseen data.

EVALUATION



The confusion matrix indicates that the model has a tendency to be more False Positive in its predictions, tending to classify messages as “Scam“ more often than they actually are.

The model’s performance was assessed on the test dataset with various metrics: Accuracy, Precision, Recall, F1-score, and Confusion Matrix to breakdown the correct and incorrect predictions

Metrics	Result
Accuracy	0.72
Precision	0.72
Recall	0.72
F1-Score	0.72

INFERENCE

`The inference process predicts probabilities for “Scam“ (1) and “Non-Scam“ (0) classes through these steps:

- Tokenization: Converts input text into token IDs.
- Model Forward Pass: Processes tokens through the pre-trained GPT model.
- Classification: The custom head predicts the class.
- Output: Returns the predicted class and its probability.

To enable deployment, the trained model and tokenizer were saved to a directory, allowing easy reloading for classifying unseen Thai SMS. An example demonstrates its accuracy in detecting a scam SMS.


```
Single Text Prediction:
Text: ยินดีด้วยคุณได้รับรางวัลจากธนาคาร กรุณาโอนค่าธรรมเนียม 1,000 บาท
Prediction: Scam
Confidence: 94.22%
Class Probabilities: {'Non-scam': 0.057820286601781845, 'Scam': 0.9421797394752502}
```

THANK YOU!

- Siriwan Sreebutkot 6520422019
- Chalita Iamleelaporn 6610412002
- Ranakorn Boonsuankergchai 6610412003