

DEPLOYMENT_README.md: BeSafe Threat Detection Model Specs (ONNX Version)

This document outlines the required files, constants, and preprocessing steps necessary to integrate the ONNX model into the mobile application. The native code (e.g., Swift, Kotlin) must replicate these steps exactly.

1. Required Files for Integration

The developer must bundle the following three files with the mobile application:

File Name	Extension	Contents	Role in Application
SafeHer_ThreatModel	.onnx	The Trained BiLSTM Network.	The core model executable for prediction via the ONNX Runtime.
SafeHer_Vocabulary	.csv	The Word-to-Index Map.	A plain-text lookup table for converting words to required integer indices.
DEPLOYMENT_README	.md or .txt	This Specification Summary.	Defines required constants and output mapping.

2. Preprocessing Pipeline Requirements (Critical)

The app's native code must replicate the following preprocessing steps to transform raw user text into the correct input tensor for the ONNX model.

A. Preprocessing Constants

Requirement	Value	Description
Input Sequence Length	200	Every input sequence (tensor) must be padded or truncated to this fixed size.
Padding/OOV Value	0 (Integer)	Used to fill sequences shorter than 200, AND used as the index for Out-Of-Vocabulary (OOV) words (words not found in the CSV map).

B. Step-by-Step Logic

- Tokenization: Split the raw input text into individual words (tokens).
- Vocabulary Lookup: For each word token:
 - Look up the word in the SafeHer_Vocabulary.csv map.
 - If the word is found, use the corresponding 0-based integer index (from the CSV file).
 - If the word is not found, use the Padding/OOV Value (0) as its index.

- **Sequence Creation:** Collect the resulting list of integers.
- **Padding/Truncation:** Adjust the sequence length to exactly 200. Crucially, the padding/truncation location used during training must be maintained:
 - If Padding was Pre-Padding (at the start of the sequence): Pad the front of the sequence with Os.
 - If Padding was Post-Padding (at the end of the sequence): Pad the end of the sequence with Os.
- **ONNX Tensor Format:** The final 200-element integer sequence must be formatted as the correct input tensor shape required by the ONNX runtime (usually [1, 200] or [200, 1]).

3. Output Classification Mapping (8 Classes)

The ONNX model returns an array of 8 scores (probabilities) in a fixed order. The developer must use the following table to interpret the output.

Output Index (0-Based)	Sanitized Name (Recommended for UI/Logging)	Original Class Name
0	I_llReportYouImmediately_	'I'll report you immediately!'''
1	I_mWarningYou_	'I'm warning you!'''
2	Non_Threat	'Non-Threat'
3	Threat	'Threat'
4	help_	'help!'''
5	pleaseHelpQuickly_	'please help quickly!'''
6	shoutingThreats_	'shouting threats!'''
7	someonelsTryingToHarmMe_	'someone is trying to harm me!'''

The final prediction should be the category corresponding to the index with the highest score.

4. Developer Confirmation

The developer needs to confirm the padding direction (pre-padding or post-padding) used in the original MATLAB pipeline to ensure input sequences are aligned correctly for the BiLSTM layers