ML4CRYPTO CHALLENGE

Team Members

Mukul Mundle

Arpan Singh

Monal Goel

Divyansh Kasture

Kaustubh Ranade

From PGDBA IIM Calcutta, ISI Kolkata, and IIT Kharagpur

Problem Statement

- The Challenge is about performing distinguishing attack on some popular symmetric-key block based ciphers with machine learning.
- The attack is a multi class classification with labels 0,1,2,3.
- Each label basically represent an algorithm of encryption ex: KASUMI, AES-128, RSA algorithms.
- Mainly 3 feature engineering techniques available for cipher blocks published by NIST.
- We use the ‘frequency within blocks’ method for FE.

Algorithm Used for Classify the attack on symmetric key block based ciphers

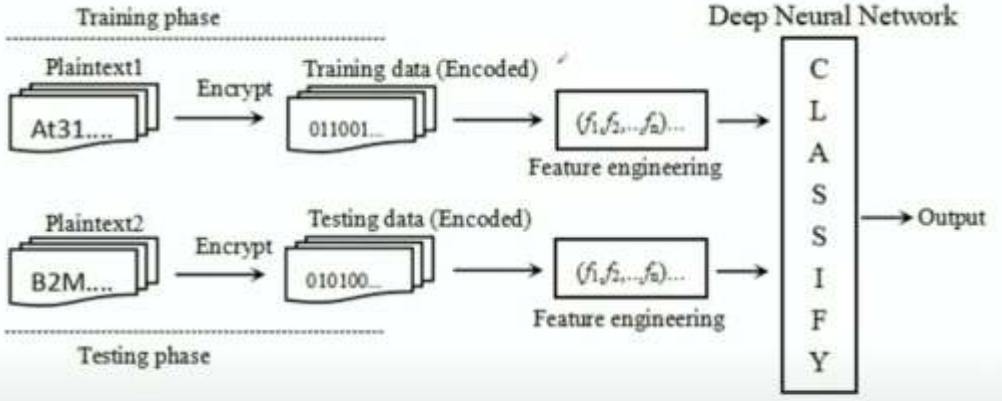
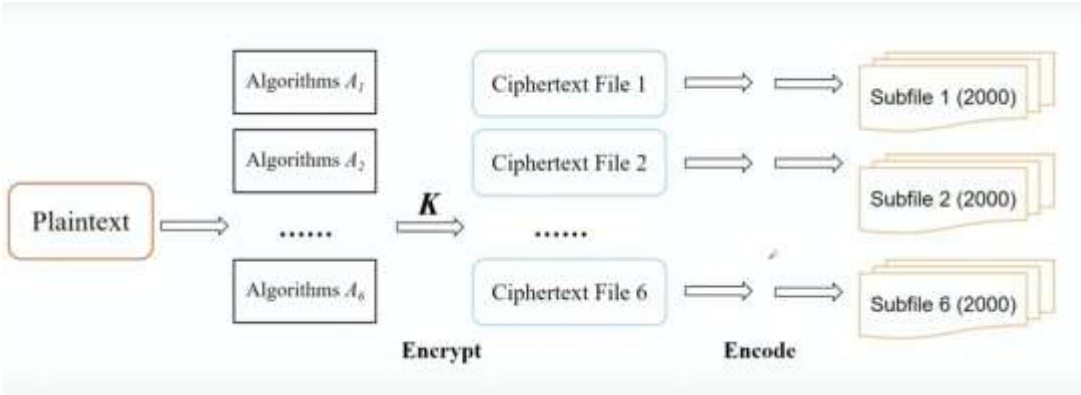
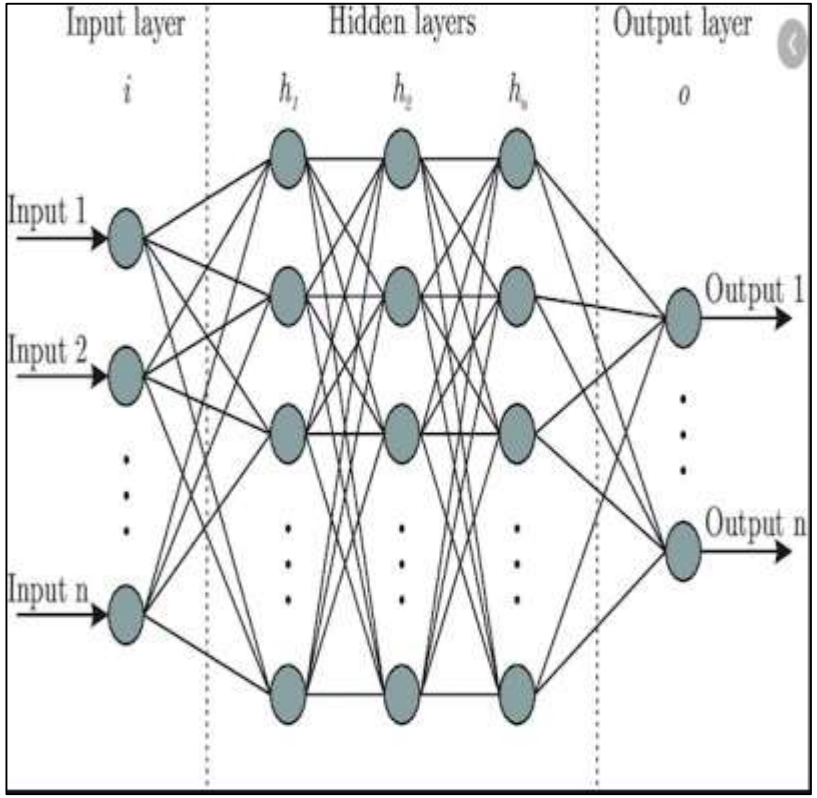


Fig 1. Cryptographic algorithms identification model



Our Approach to Solve the Problem

1. Converting the Plaintext which is in Hexadecimal format to Binary format. Dataset is having 80000 Rows with 1 Target variable having 4 classes.
2. After encrypting the dataset in the binary form we have done feature extraction from the long binary text.
3. We have done pattern recognition from the long binary text.
4. Identified patterns frequency in each binary text - '010', '0110', '01110' etc.
5. After identifying pattern frequency for different distribution('010', '0110', '01110')
6. Further we are dividing our dataset into training and testing datasets.
7. As a model building we have used a deep neural network on the training dataset.



1st Approach:

- We have used a Deep neural network with 4 hidden layers. We have created 22 patterns which are basically created features for our neural network model.
- We have used the Relu function as an Activation function and ADAM for optimization.
- We have iterated the entire model for 200 epochs with a batch of 5.
- We have calculated cross-entropy loss for the test dataset ~1.3.
- Final accuracy of the deep neural network model is ~25%.
- We have used the Adaboost model as well which provides a similar accuracy of ~25% for the test dataset.

2nd Approach:

- Used BertTokenizer from transformers library to tokenize the text based on pretrained hugging face model 'bert-base-cased'
- Generated set of tokens and encoding matrix for every cipher text.
- Used two input layers and attention masks for our model model using tensorflow.
- Achieved an accuracy of around 26%.

The 2nd Approach model was selected as our final submission model.