CS771 Assignment 1
Team: Neural Ninjas 2

Group Members:

Avinash Shukla - 210236
Chirumamilla Satya Keerthana - 210290
Mantapuram Shreeja - 210592
Sampada Kalavakunta - 210914
Yerusu Dharini Reddy - 211204

1 Problem: Cracking the CAR PUF

A CAR PUF has two PUFs: a working PUF and a reference PUF, both fed with the same 32-bit challenge.

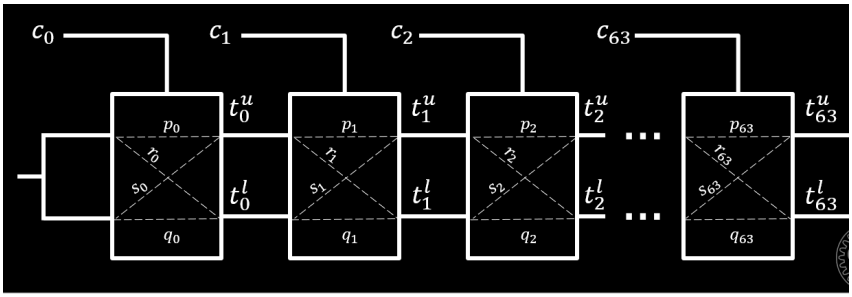

Figure 1: PUF with 64 bit input, Reference: Lecture Slides

As derived in class, for an arbitrary PUF with 32 input bits, we have:

$$\Delta = w_0x_0 + w_1x_1 + w_2x_2 + \ldots + w_{31}x_{31},$$

Δ is the time difference between the upper and lower signals, where :

$$x_i = d_id_{i+1}\ldots d_{31}, \quad d_i = 1 - 2c_i,$$

and

$$w_0 = \alpha_0, \quad w_i = \alpha_i + \beta_{i-1} \quad \text{for } i > 0.$$

$$\alpha_i = (p_i - q_i + r_i - s_i)/2, \quad \beta_i = (p_i - q_i - r_i + s_i)/2,$$

where p_i, q_i, r_i, s_i are the delay introduced by the i^{th} MUX.

For the CAR PUF in our problem:

Δ_w is the time difference of the working PUF and Δ_r is the time difference of the reference PUF.

$$\Delta_w = w_0x_0 + w_1x_1 + w_2x_2 + \ldots + w_{31}x_{31} + \beta_{31},$$

$$\Delta_r = w_0^{'}x_0 + w_1^{'}x_1 + w_2^{'}x_2 + \ldots + w_{31}^{'}x_{31} + \beta_{31}^{'}.$$

It can also be written in vector form as:

$$\Delta_w = U^T x + p, \quad \Delta_r = V^T x + q.$$

It is given that if $|\Delta_w - \Delta_r| \leq \tau$, then response = 0 and if $|\Delta_w - \Delta_r| > \tau$, then response = 1.

We square the inequality so that we can train the model independent of τ. We do this so that we can write it in the form of a sign function as shown:

$$|\Delta_w - \Delta_r|^2 - \tau^2 \leq 0, \text{ response = 0}$$
$$|\Delta_w - \Delta_r|^2 - \tau^2 > 0, \text{ response = 1}$$

$$\text{Hence, response} = \frac{1 + \text{sign}(|\Delta_w - \Delta_r|^2 - \tau^2)}{2}.$$

Subtracting the two equations we get:

$$\Delta_w - \Delta_r = (w_0 - w_0^{'})x_0 + (w_1 - w_1^{'})x_1 + \ldots + (w_{31} - w_{31}^{'})x_{31} + \beta_{31} - \beta_{31}^{'}.$$

Let $w_i - w_i^{'} = A_i$ for $i = 0, 1, \ldots, 31$ and $\beta_{31} - \beta_{31}^{'} = \beta$.

$$\Delta_w - \Delta_r = A_0 x_0 + A_1 x_1 + \ldots + A_{31} x_{31} + \beta.$$

$|\Delta_w - \Delta_r|^2 - \tau^2 = A_0^2 x_0^2 + A_1^2 x_1^2 + \ldots + A_{31}^2 x_{31}^2 + 2\sum_{i=0}^{31} \sum_{j=1,j>i}^{31} A_i A_j x_i x_j + 2\beta \sum_{i=0}^{31} A_i x_i + \beta^2 - \tau^2$.

Since $x_i = d_i d_{i+1} \ldots d_{31}$ and $d_i \in \{-1, 1\}$, therefore $x_i^2 = 1$ for all $i = 0, 1, 2, \ldots, 31$.

Simplifying the expression, we get:

$$|\Delta_w - \Delta_r|^2 - \tau^2 = 2\beta \sum_{i=0}^{31} A_i x_i + 2 \sum_{i=0}^{31} \sum_{j=1,j>i}^{31} A_i A_j x_i x_j + b,$$

where $b = \sum_{i=0}^{31} A_i^2 + \beta^2 - \tau^2$.

To achieve the final linear model:

Let the feature vector be $\phi(x) = [x_0, x_1, x_2, \ldots, x_{31}, x_0 x_1, x_0 x_2, x_0 x_3, \ldots, \text{all } x_i x_j \text{ terms}]$.

Let $W^T = [2\beta A_0, 2\beta A_1, \ldots, 2\beta A_{31}, 2A_0 A_1, 2A_0 A_2, \ldots, 2A_{30} A_{31}]$.

In the feature vector, we have 32 x_i terms and $\binom{32}{2}$ $x_i x_j$ terms.

Therefore, the dimensionality D of $\phi(x) = \binom{32}{2} + 32 = 528$.

Rewriting in vector form: $|\Delta_w - \Delta_r|^2 - \tau^2 = W^T x + b$ (can be expressed as a linear model)

Thus, there exists $\mathbf{w} \in \mathbb{R}^{528}$, $b \in \mathbb{R}$ such that for all challenges $c \in \{0, 1\}^{32}$, the following expression:

$$\frac{1 + \text{sign}(w^T \phi(c) + b)}{2}$$

gives the correct response.

2 Problem 2: Code is in the submitted file

3 Problem 3

3.1 Loss hyperparameter in LinearSVC

The loss function characterizes how well the model performs on a given dataset. `sklearn.svm.LinearSVC` provides two loss functions, namely: 'hinge' and 'squared_hinge'.

Loss function	Training Time	Accuracy
'hinge' loss	9.87	98.872
'squared_hinge' loss	10.04	99.124

Table 1: Effect of loss functions on LinearSVC Model

3.2 C in LinearSVC and Logistic Regression

C value	Training Time (SVC)	Accuracy (SVC)	Training Time (Logistic)	Accuracy (Logistic)
0.001	1.85	96.42	1.20	90.69
0.01	5.48	98.65	1.49	96.35
0.1	13.54	98.99	1.701	98.71
1	10.23	99.14	1.875	99.07
10	9.20	99.13	8	99.29
100	9.47	99.26	2.14	99.36
1000	10.05	98.87	1.25	99.33

Table 2: Effect of C values on training time and accuracy

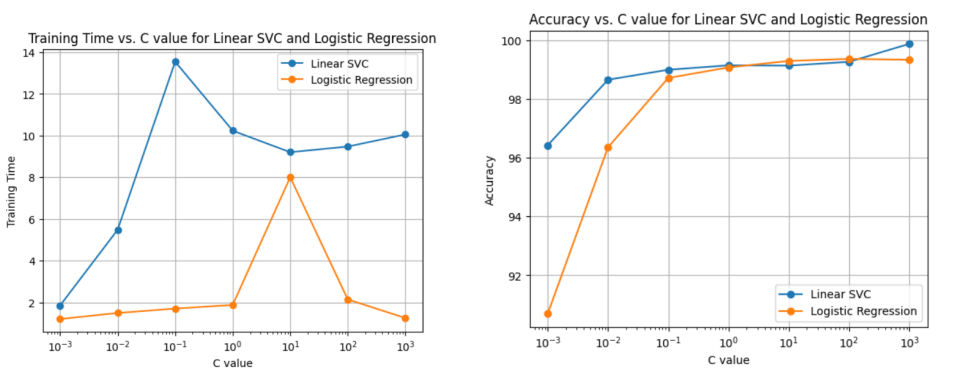

Figure 2: Training Time and Accuracy vs C value

Tolerance	Training Time (SVC)	Accuracy (SVC)	Training Time (Logistic)	Accuracy (Logistic)
10^{-7}	9.95	99.13	2.94	99.05
10^{-6}	10.14	99.20	2.81	99.06
10^{-5}	9.79	99.11	2.77	99.05
10^{-4}	9.98	99.21	2.74	99.06
10^{-3}	9.93	99.13	2.75	99.06
10^{-2}	10.35	99.14	2.85	99.06
10^{-1}	9.85	99.12	2.60	99.07
1^{-1}	10.30	99.23	2.49	99.07

Table 3: Effect of tolerance values on training time and accuracy

Figure 3: Accuracy and Training Time vs Tolerance

Also, we are finally using LogisticRegression Model out of all models in our final code because it has a faster training time without compromising on accuracy. In the final code, we have not used any hyperparamters so that the code is faster and has similar accuracy when used with hyperparameters.