CS771 Assignment 1
Team 82 - Syntax_Error

Team Members
Aryan Jain - 200201
Ritik Raj - 200803
Shivang Pandey - 200941
Varun Maghnani - 201092
Suhani - 201011
Kashishpreet Kaur - 200496

1 Modelling a simple XORRO PUF

For a single XORRO let us consider two cycles of oscillation - one having output 1, other having output 0. Given that all the other inputs $(a_0, a_1, a_2, ...a_{R-1})$ are same in both the oscillations, the output across each XOR gate would be different across both the cycles. The delay across i^{th} XOR gate with inputs 1 and a_i would be $(1-a_i)\delta_{10}^{ui}+a_i\delta_{11}^{ui}$, and for inputs 0 and a_i would be $(1-a_i)\delta_{00}^{ui}+a_i\delta_{01}^{ui}$. Thus the delay for the upper XORRO across the ith XOR gate for both cycles would be

$$\Delta_i^u = (1-a_i)\delta_{10}^{ui} + a_i\delta_{11}^{ui} + (1-a_i)\delta_{00}^{ui} + a_i\delta_{01}^{ui}$$

similarly for lower XORRO,

$$\Delta_i^l = (1-a_i)\delta_{10}^{li} + a_i\delta_{11}^{li} + (1-a_i)\delta_{00}^{li} + a_i\delta_{01}^{li}$$

The total delay across upper and lower XORROs would be

$$\Delta^u = \sum_{i=0}^{R-1}[(1-a_i)(\delta_{10}^{ui} + \delta_{00}^{ui}) + a_i(\delta_{11}^{ui} + \delta_{01}^{ui})]$$

$$\Delta^l = \sum_{i=0}^{R-1}[(1-a_i)(\delta_{10}^{li} + \delta_{00}^{li}) + a_i(\delta_{11}^{li} + \delta_{01}^{li})]$$

The output of the XORRO PUF depends on the time difference between the total time delay across the XORROs

$$\Delta = \sum_{i=0}^{R-1}[(1-a_i)(\delta_{10}^{ui} + \delta_{00}^{ui} - \delta_{10}^{li} - \delta_{00}^{li}) + a_i(\delta_{11}^{ui} + \delta_{01}^{ui} - \delta_{11}^{li} - \delta_{01}^{li})]$$

substituting $\alpha_i = \delta_{10}^{ui} + \delta_{00}^{ui} - \delta_{10}^{li} - \delta_{00}^{li}$, and $\beta_i = \delta_{11}^{ui} + \delta_{01}^{ui} - \delta_{11}^{li} - \delta_{01}^{li}$,

$$\Delta = \sum_{i=0}^{R-1}[(1-a_i)\alpha_i + a_i\beta_i]$$

$$= \sum_{i=0}^{R-1}\alpha_i + \sum_{i=0}^{R-1}a_i(\beta_i - \alpha_i)$$

assuming $b = \sum_{i=0}^{R-1}\alpha_i$, $\mathbf{w} = \{\beta_0 - \alpha_0, \beta_1 - \alpha_1, ...\beta_{R-1} - \alpha_{R-1}\}$ and $\phi(\mathbf{c}) = \{a_0, a_1, ...a_{R-1}\}$. Now, the ouput of the simple XORRO PUF is 1 if the upper XORRO has higher frequency, 0 if lower

Preprint. Under review.

XORRO has higher frequency. Higher frequency means lower time delay, thus the response depends on the sign of Δ, and can be expressed as,

$$response = \frac{1 + sign(\Delta)}{2}$$
$$response = \frac{1 + sign(\mathbf{w}^T \phi(\mathbf{c}) + b)}{2}$$

2 Extending the Linear Model to crack an Advanced XORRO PUF

The Extended XORRO PUF with Multiplexers is an improved version of the XORRO PUF which was designed to be stronger and more secure. The previous version of the XORRO PUF had just 2 XORROs and no select bits or multiplexers. However, in the Extended XORRO PUF, the number of XORROs is increased to 2^S, and select bits are added to the challenge vector. The select bits are used to choose the two XORROs that are compared to produce the response. The select bits are divided into two groups: the first S select bits are used to select the first XORRO and the second S select bits are used to select the second XORRO. The selection is done using a multiplexer. For example, if S is equal to 3, there are 8 XORROs (2^3), and the 2^S bits are 110010, then XORRO number 6 will be selected as the first XORRO ($110 \equiv 6$), and XORRO number 2 will be selected as the second XORRO ($010 \equiv 2$). As a result, we developed M = $2^{(s-1)}(2^s - 1)$ (where s = 4) different models, marked them according to their numbers, and with the assistance of 2s select bits, we have selected the upper and lower models and calculated their frequencies. If the frequency of the upper model is greater than the lower model, the output will be 1, otherwise it will be 0.

3 Results obtained from the code

- Training time = 1.73 sec
- Testing time = 2.21 sec
- Size = 85233.0
- Accuracy = 94.744 %

4 Effect of various hyperparameters on training time and test accuracy

The following are the outcomes of experiments with both the sklearn.svm.LinearSVC and sklearn.linear model.LogisticRegression methods when used to learn the ensemble linear model.

All other parameters (other than the ones that are changed) are the default parameters.

4.1 Changing the loss hyperparameter in LinearSVC (hinge vs squared hinge)

Loss Hyperparameter	Training time (in sec)	Test accuracy (in %)
Hinge	2.31	93.953
Square hinge	2.12	94.743

4.2 Setting C hyperparameter in LinearSVC and LogisticRegression to high/low/medium values

For LinearSVC

C Hyperparameter	Training time (in sec)	Test accuracy (in %)
0.01	1.11	90.397
0.1	1.33	93.998
0.5	2.32	94.67
1	2.83	94.742
10	2.44	94.268
100	2.31	93.885
1000	2.32	93.849

For Logistic Regression

C Hyperparameter	Training time (in sec)	Test accuracy (in %)
0.01	2.20	82.517
0.1	2.27	89.633
0.5	2.67	93.097
1	2.99	93.917
10	4.09	94.947
100	5.57	94.943
1000	5.94	94.743

4.3 Changing the penalty (regularization) hyperparameter in LinearSVC and LogisticRegression (l2 vs l1)

For LinearSVC

Penalty Hyperparameter	Training time (in sec)	Test accuracy (in %)
l1	10.45	94.596
l2	2.78	94.739

For Logistic Regression

Penalty Hyperparameter	Training time (in sec)	Test accuracy (in %)
l1	2.61	94.452
l2	3.01	93.917

References

[1] www.geeksforgeeks.org

[2] www.stackoverflow.com