CS771 Assignment 1
Team: Overfitted Models

Group Members:

Aravind Sheshadri - 210180
K N Joshua - 210489
Raghav Shukla - 210800
Sai Praneeth Donthu - 210900
Shobhit Sharma - 210992
Sumay Avi - 211071

1 Companion Arbiter PUF

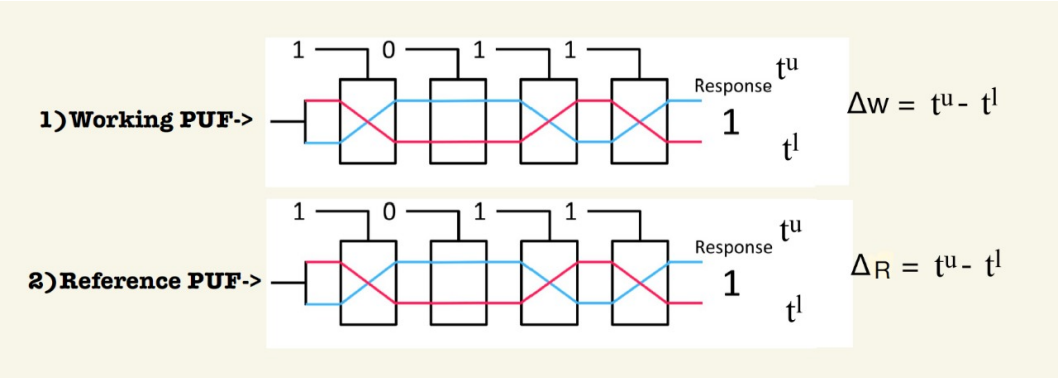

Figure 1: Companion Arbiter PUF

Let Δ_w be the difference in timings of the Working PUF and Δ_r be the difference in timings of the Reference PUF. The overall response of the CAR PUF, $r \in \{0, 1\}$, is then as follows:

$$r = \left\{ \begin{array}{ll} 0, & \text{if } |\Delta_w - \Delta_r| \leq \tau \\ 1, & \text{if } |\Delta_w - \Delta_r| > \tau \end{array} \right\} \tag{1}$$

Where τ is the threshold value such that $\tau > 0$
We need to show that there exists a mapping $\phi : \{0, 1\}^{32} \rightarrow \mathbb{R}^D$, $D > 0$, a D - dimensional model $\mathbf{W} \in \mathbb{R}^D$ and a bias term $b \in \mathbb{R}$ such that the CAR PUF can be simplified into a linear model $\mathbf{W}^T \phi(\mathbf{c}) + b$ such that for all challenge pairs $(\mathbf{c}, r)$, $\mathbf{c} \in \{0, 1\}^{32}$, $r \in \{0, 1\}$

$$r = \frac{1 + \text{sign}(\mathbf{W}^T \phi(\mathbf{c}) + b)}{2} \tag{2}$$

2 Linear Model of a PUF

For an individual PUF consisting of a chain of n multiplexers, the model derived is linear and in the form:

$$\Delta = \mathbf{W}^T x + b \tag{3}$$

Preprint. Under review.

where,
W is the feature vector, $\mathbf{W} \in \mathbb{R}^n$
x is the encoded challenge response pairs, $x = [x_1, x_2, \ldots, x_n] \in \{-1, 1\}^n$, such that

$$x_i = (1 - 2c_i) \times (1 - 2c_{i+1}) \times \cdots \times (1 - 2c_n) \tag{4}$$

and b is the bias term, $b \in \mathbb{R}$.
The working and reference PUF's are given by models say Δ_w and Δ_r respectively where,

$$\Delta_w = \mathbf{U}^T x + p = \tilde{\mathbf{U}}^T \tilde{x}$$
$$\Delta_r = \mathbf{V}^T x + q = \tilde{\mathbf{V}}^T \tilde{x}$$

3 Companion PUF

For the Companion PUF we have frome equation (1),

$$r = \frac{1 + \text{sign}(|\Delta_w - \Delta_r| - \tau)}{2} \tag{5}$$

since $\tau \geq 0$ we can rewrite in the following manner,

$$\begin{aligned}\text{sign}(|\Delta_w - \Delta_r| - \tau) &= \text{sign}((\Delta_w - \Delta_r)^2 - \tau^2) \\ &= \text{sign}(((\tilde{\mathbf{U}} - \tilde{\mathbf{V}})^T x)^2 - \tau^2) \\ &= \text{sign}((\tilde{\mathbf{A}}^T x)^2 - \tau^2)\end{aligned}$$

where, $\tilde{\mathbf{A}} = \tilde{\mathbf{U}} + \tilde{\mathbf{V}}$, further we have,

$$\begin{aligned}&= \text{sign}((\sum_{i=1}^{33} A_i x_i)^2 - \tau^2) \\ &= \text{sign}(\sum_{i=1}^{33} A_i A_j x_i x_j - \tau^2) \\ &= \text{sign}(\sum_{i=1, i=j}^{33} A_i^2 x_i^2 + \sum_{i=1, i\neq j}^{33} A_i A_j x_i x_j - \tau^2)\end{aligned} \tag{6}$$

To arrive at a linear model, we make the following changes
$\mathbf{W}^T = \left[A_1^2, A_2^2, \ldots, A_{33}^2, A_1.A_2, A_1.A_3, \ldots, A_{32}.A_{33}\right]_{1\times 1089}$
$\phi(c) = \left[x_1^2, x_2^2, \ldots x_{33}^2, x_1.x_2, x_1.x_3, \ldots, x_{32}.x_{33}\right]_{1089\times 1}$
$b = -\tau^2$
where, c = $[x_1, x_2, x_3, \ldots, x_{32}, 1]_{33\times 1}$
$c_{33} = x_{33} = 1$, due to the bias term

Since $x_i \in \{-1, 1\}$ we have that $x_i^2 = 1$
Thus, these terms are then accounted for in the bias term.
Now equation 6 can be finally rewritten as

$$= \text{sign}(\mathbf{W}^T \phi(\mathbf{c}) + b)$$

For the $i \neq j$ terms: $x_i x_j = x_j x_i$

$$\begin{aligned}\implies \text{sign}(\mathbf{W}^T \phi(\mathbf{c}) + b) &= \text{sign}(\sum_{i=1, i=j}^{33} A_i^2 x_i^2 + \sum_{i=1, i\neq j}^{33} A_i A_j x_i x_j - \tau^2) \\ &= \text{sign}(2 \times \sum_{i=1, i<j}^{33} A_i A_j x_i x_j + \sum_{i=1, i=j}^{33} A_i^2 - \tau^2)\end{aligned}$$

There is no reason to consider all the elements since they get repeated hence we just take half of the elements.
Thus, total number of elements in $\phi(c) = \frac{1089-33}{2} = 528$
$\implies$ Dimensionality of $\phi(c)$, D = 528

3.1 Final Model

Hence the derived final model is:-

$$r = \frac{1 + \text{sign}(\mathbf{W}^T \phi(\mathbf{c}) + b)}{2} \tag{7}$$

where, $\mathbf{W}^T = \left[[A_i.A_j]_{i \neq j}\right]_{1 \times 528}$

$\phi(c) = \left[[x_i.x_j]_{i \neq j}\right]_{528 \times 1}$

$b = \sum_{i=1}^{33} A_i^2 - \tau^2$

4 Observations

4.1 Hinge vs Squared Hinge in LinearSVC

Type of Loss	Training Time (Linear)	Accuracy
Hinge	8.915139170199996	98.85
Squared Hinge	9.489233391199946	99.122

Table 1: Hinge vs Squared Hinge in LinearSVC

It can be seen that as we shift the type of loss from hinge loss to squared hinge loss, the training time has a slight increase (almost negligible). This also results in a slight increase in accuracy as well as seen by the numbers above.

4.2 Varying C value in LinearSVC and LogisticRegression

C Value	Training Time (Linear)	Accuracy (Linear)	Training Time (Logistic)	Accuracy (Logistic)
0.001	2.780143820000012	95.97	1.6377086851999991	90.69
0.01	4.527520865600081	98.65	1.7146549878000088	96.35
0.1	11.580320310199932	98.99	1.8066378736000162	98.71
1	9.886037395999939	99.15	1.9118000148000192	99.07
10	9.40738098439997	99.032	2.140517579800007	99.22
100	9.59281371779998	99.004	2.577377421199981	99.31
1000	10.771529467599976	98.97	3.3807954380000185	99.23

Table 2: Varying C value in LinearSVC and LogisticRegression

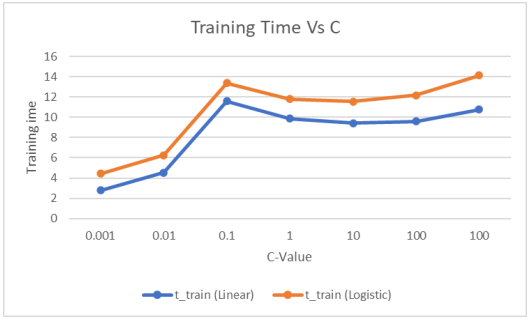

Figure 2: Training Time v/s C value

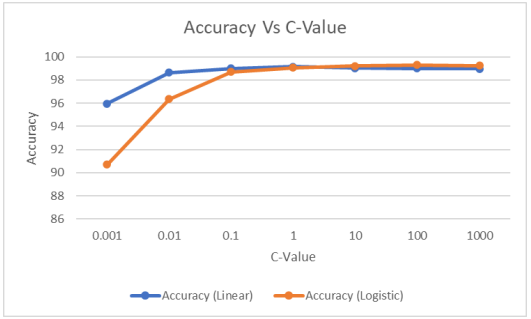

Figure 3: Accuracy v/s C value

We are using logistic without hyperparameters tuning which gives fastest training without loss of efficiency.
It can firstly be seen that the training times required for Logistic Regression are higher than that required for the Linear SVM Model. The training times for both SVM and Logistic Regression first

increase till C=0.1 then decrease. Also, at lower C values (C < 1): it can also be seen that the accuracy of the Linear SVM Model is higher than that of Logistic Regression. However, as the value of C increases, it can be seen that the accuracy of the SVM Model as well as that of Logistic Regression slowly decreases (nearly constant).

4.3 Varying Tolerance in LinearSVC and LogisticRegression

Tolerance	Training Time (Linear)	Accuracy (Linear)	Training Time (Logistic)	Accuracy (Logistic)
10^{-7}	9.934231497599649	99.142	2.9431920086000902	99.06
10^{-6}	10.165963900399948	99.116	2.8188486281999756	99.06
10^{-5}	9.890949846599687	99.092	2.7918271808001007	99.06
10^{-4}	9.974957427400113	99.138	2.7290737153998634	99.07
10^{-3}	9.9194905136002487	99.124	2.729843824600084	99.07
10^{-2}	10.125141818599877	99.129	2.884505387199897	99.07
10^{-1}	9.760532332399999	99.107	2.6047298403999775	99.07
1	10.21900592059992	99.149	2.5136472201998914	99.07

Table 3: Varying Tolerance value in LinearSVC and LogisticRegression

We notice that LinearSVM's model training time and accuracy barely changes as the tolerance is increased. This is because it hits the limit on the number of maximum iterations allowed before it reaches the tolerance. Thus, there is no change in the values. The same can be said for the Logistic Regression Model, whose accuracy does not change as the tolerance increases, but its training time keeps reducing as tolerance keeps increasing.