CS771 Assignment-1

Padulkar Rohan Ravikumar
210689
rohan21@iitk.ac.in

Harshit Shakya
210427
harshit21@iitk.ac.in

Suvrat Pal
211089
suvratp21@iitk.ac.in

Utkarsh Mishra
211131
utkarshm21@iitk.ac.in

Shishir Gujarey
210977
shishirg21@iitk.ac.in

Aakash Yadav
210010
aakashy21@iitk.ac.in

1 Part 1

For a single arbiter PUF, we know:

$$\Delta = \mathbf{w}^T\mathbf{x} + b$$

where $w_0 = \alpha_0$, $w_i = \alpha_i + \beta_{i-1}$ (for $i > 0$) and α_i, β_i account for the delays incurred in the individual mux
and x is defined by the map $\zeta : \mathbf{c} \to \mathbf{x}$ (for the case of 32-bit challenges) such that:

$$\mathbf{x} \triangleq \begin{bmatrix} (1-2c_0)(1-2c_1)\dots(1-2c_{31}) \\ \vdots \\ (1-2c_{31}) \end{bmatrix}_{32} \tag{1}$$

where c_i are the challenge bits

Thus, for the CAR-PUF, we have

$$\Delta_w = \mathbf{u}^T\mathbf{x} + p$$
$$\Delta_r = \mathbf{v}^T\mathbf{x} + q$$

where Δ_w and Δ_r are the difference in timings experienced by the working and reference PUFs respectively for the same challenge.

According to the question, The response is 0 if $|\Delta_w - \Delta_r| \leq \tau$ and 1 otherwise. Let us consider the case where the response should be 0. Squaring both sides to handle mod:

$$\Rightarrow (\Delta_w - \Delta_r)^2 \leq \tau^2$$

$$\Rightarrow \left[(\mathbf{u} - \mathbf{v})^T\mathbf{x} + (p-q)\right]^2 < \tau^2$$

$$\Rightarrow (\mathbf{w}^T\mathbf{x} + b)^2 \leq \tau^2$$

where $\mathbf{w} \triangleq \mathbf{u} - \mathbf{v}$ and $b \triangleq p - q$. Thus,

$$\Rightarrow (\mathbf{w}^T\mathbf{x})^2 + b^2 + 2(\mathbf{w}^T\mathbf{x})b \leq \tau^2$$

$$\Rightarrow \left(\sum_{i=0}^{31} w_i x_i\right)^2 + 2(\mathbf{w}^T\mathbf{x})b + (b^2 - \tau^2) \leq 0$$

$$\Rightarrow \sum_{i=0}^{31}(w_i x_i)^2 + 2\sum_{i=0}^{31}\sum_{j=i+1}^{31} w_i w_j x_i x_j + 2(\mathbf{w}^T\mathbf{x})b + (b^2 - \tau^2) \leq 0$$

$$\Rightarrow \sum_{i=0}^{31}(w_i x_i)^2 + 2\sum_{i=0}^{31}\sum_{j=i+1}^{31} w_i w_j x_i x_j + 2\left(\sum_{i=0}^{31} w_i x_i\right) b + (b^2 - \tau^2) \leq 0$$

Since $c_i \in \{0, 1\}$, $x_i = 1 - 2c_i = \pm 1 \implies x_i^2 = 1$. Hence, $\sum\limits_{i=0}^{31}(w_i x_i)^2 = \sum\limits_{i=0}^{31} w_i^2$, which is a constant for 2 fixed PUFs

$$\Rightarrow 2\sum_{i=0}^{31}\sum_{j=i+1}^{31} w_i w_j x_i x_j + 2\left(\sum_{i=0}^{31} w_i x_i\right) b + (b^2 + \sum_{i=0}^{31} w_i^2 - \tau^2) \leq 0 \tag{2}$$

In the above expression, the first summation consists of $\binom{32}{2}$ = 496 terms, the second summation consists of 32 terms and the rest are constants:

Now, let ψ be the map $\mathbf{x} \to \mathbf{X}$ such that:

$$\mathbf{X} \triangleq \begin{bmatrix} x_0 x_1 \\ \vdots \\ x_0 x_{31} \\ x_1 x_2 \\ \vdots \\ x_1 x_{31} \\ \vdots \\ \vdots \\ x_{30} x_{31} \\ x_0 \\ \vdots \\ x_{31} \end{bmatrix}_{528} \tag{3}$$

We already know a map $\zeta : \mathbf{c} \to \mathbf{x}$

Hence, we can get a map $\phi : \mathbf{c} \to \mathbf{X}$, where $\phi(c) = \psi(\zeta(c))$‘

Let $\mathbf{W}$ be the vector:

$$\mathbf{W} \triangleq \begin{bmatrix} 2w_0 w_1 \\ \vdots \\ 2w_0 w_{31} \\ 2w_1 w_2 \\ \vdots \\ 2w_1 w_{31} \\ \vdots \\ \vdots \\ 2w_{30} w_{31} \\ 2bw_0 \\ \vdots \\ 2bw_{31} \end{bmatrix}_{528} \tag{4}$$

and let $B = (b^2 + \sum w_i^2 - \tau^2)/2$

Now, we can write equation (2) in the form:

$$\mathbf{W}^T\mathbf{X} + B \leq 0 \tag{5}$$

Now, we need the response to be 0 if equation (5) is true and 1 otherwise.

Hence, we can write the response r simply as:

$$r = \frac{1 + sign(\mathbf{W}^T\mathbf{X} + B)}{2}$$

Now, if c is the challenge vector, $\mathbf{X} = \phi(c)$. Hence,

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

By definition, we can clearly see that both **X** and **W** are 528-dimensional vectors. Hence, we have found a 528-dimensional linear model which can perfectly predict the responses of a CAR-PUF.

3 Part 3

The following data shows how various hyperparameters affected training time and test accuracy.

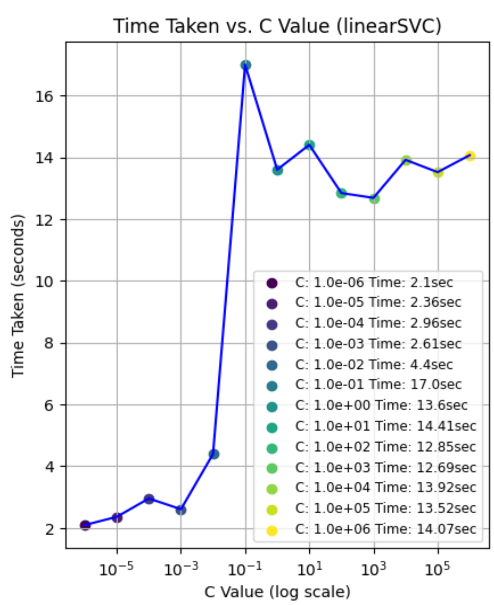
3.1 (a) Changing the loss hyperparameter in LinearSVC (Hinge vs Squared Hinge)

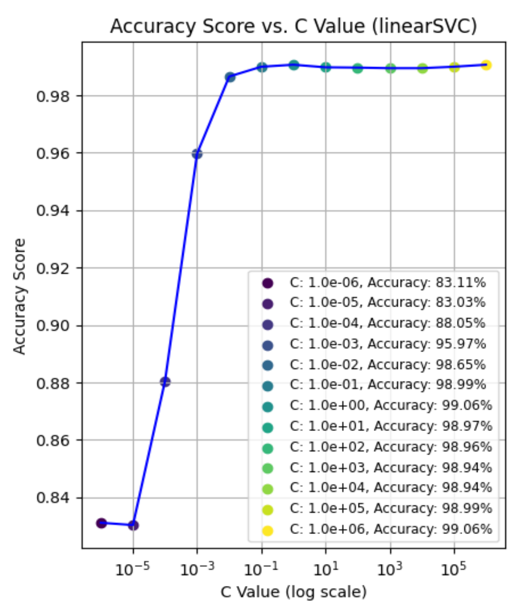
	Hinge	**Squared Hinge**
Training Time(s)	8.1138	10.6717
Test Accuracy(%)	98.88	99.02

3.2 (b) Setting Cost Parameter to high/low/medium value

For Linear SVC:

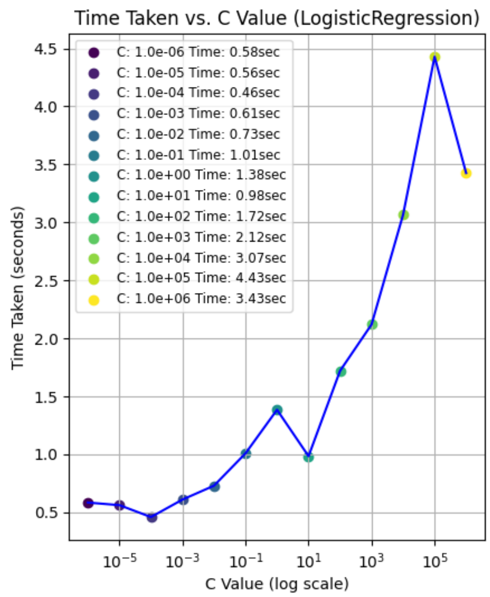
	High(C=10.0)	**Medium(C=1.0)**	**Low(C=0.1)**
Training Time(s)	8.4517	8.2536	9.9972
Test Accuracy(%)	98.95	99.12	98.99

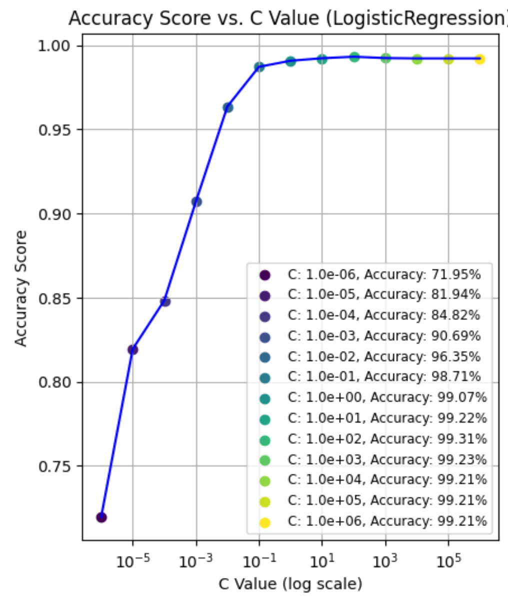




For LogisticRegression:

	High(C=10.0)	Medium(C=1.0)	Low(C=0.1)
Training Time(s)	0.8814	0.8971	0.8232
Test Accuracy(%)	99.22	99.07	98.71

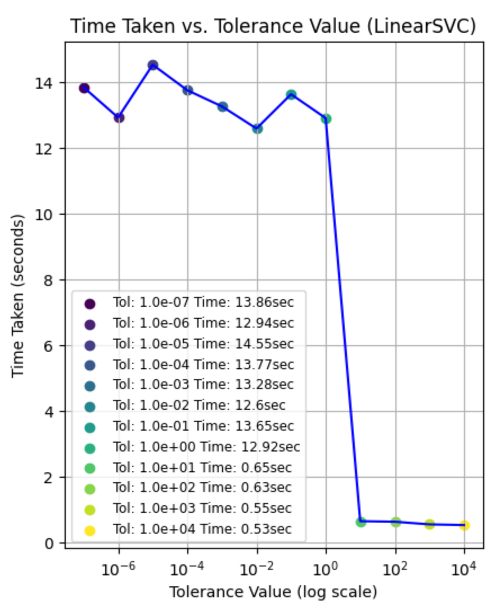


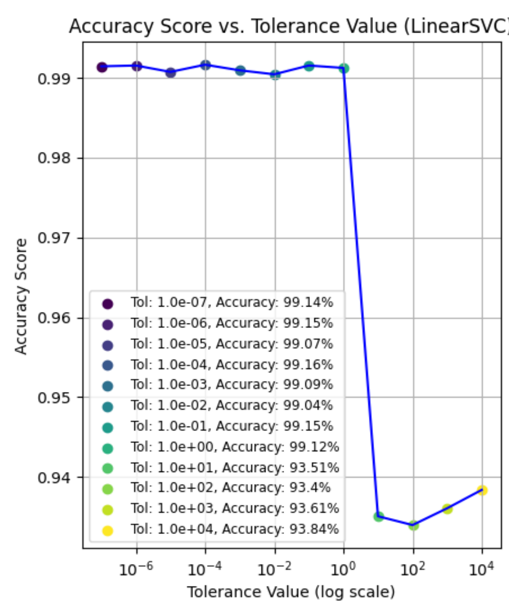


3.3 (c) Changing tolerance to high/low/medium value

For Linear SVC:

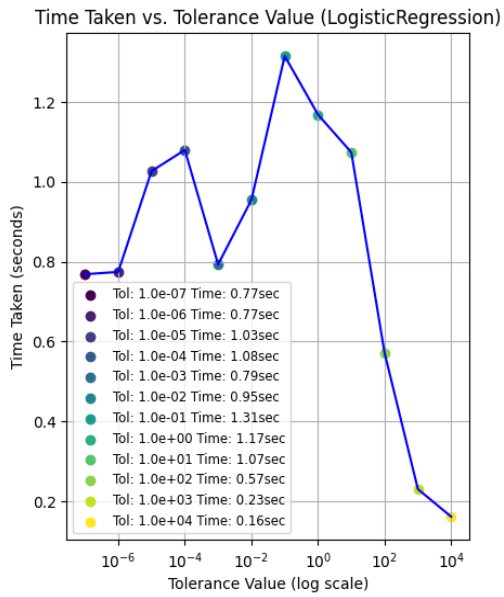
	High(tol=1e-3)	Medium(tol=1e-4)	Low(tol=1e-5)
Training Time(s)	8.5121	7.8091	8.4429
Test Accuracy(%)	99.17	99.12	99.16

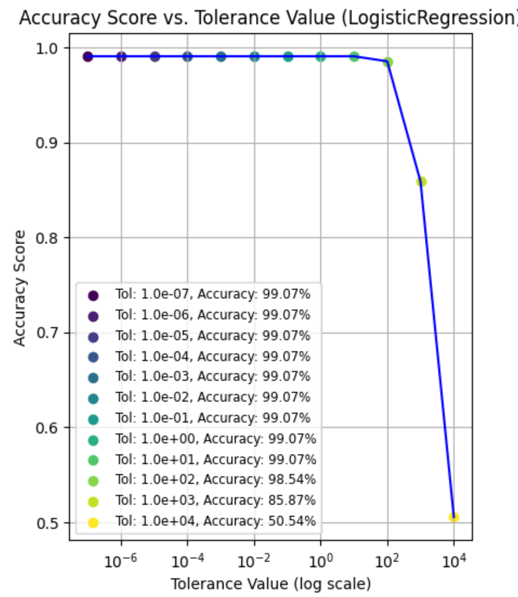




For LogisticRegression:

	High(tol=1e-3)	Medium(tol=1e-4)	Low(tol=1e-5)
Training Time(s)	0.7067	0.6907	0.7221
Test Accuracy(%)	99.07	99.07	99.07





3.4 Changing the penalty (regularization) hyperparameter (l2 vs l1)

For Linear SVC:

	l1	l2
Training Time(s)	161.6858	3.7122
Test Accuracy(%)	99.09,	99.19

For LogisticRegression (solver = 'liblinear'):

	l1	l2
Training Time(s)	202.8877	7.6563
Test Accuracy(%)	99.18	99.06