Assignment - 1

Amit Kumar Jana
210122

Anshu Saini
210156

Sarthak Agarwal
210933

Sushant Faujdar
211085

Vishant Bhadana
211177

Abstract

To prove that there does exist a linear model that can perfectly predict the responses of a Companion ArbiteR PUF.

1 Mathematical Derivation

We know, for a single PUF,

$$\Delta_{32} = w_0 \cdot x_0 + w_1 \cdot x_1 + \ldots + w_{63} \cdot x_{32} + \beta_{32} = \mathbf{w}^\top \mathbf{x} + b$$

where,

$$x_i = d_i \cdot d_{i+1} \cdot \ldots \cdot d_{32}$$

$$d_i = (1 - 2c_i); \quad c_i \text{ is the challenge in challenge vector}$$

$$w_0 = \alpha_0$$

$$w_i = \alpha_i + \beta_{i-1} \quad (for\ i > 0)$$

$$\alpha_i = \frac{p_i - q_i + r_i - s_i}{2}$$

$$\beta_i = \frac{p_i - q_i - r_i + s_i}{2}$$

In our case, given, Δ_w and Δ_r are the difference in timings experienced for the two PUFs on the same challenge, where

$$\Delta_w = u_0 \cdot x_0 + u_1 \cdot x_1 + \ldots + u_{32} \cdot x_{32} + p = \mathbf{u}^\top \mathbf{x} + p$$

$$\Delta_r = v_0 \cdot x_0 + v_1 \cdot x_1 + \ldots + v_{32} \cdot x_{32} + q = \mathbf{v}^\top \mathbf{x} + q$$

The response to this challenge is 0 if $|\Delta_w - \Delta_r| \leq \tau$ and the response is 1 if $|\Delta_w - \Delta_r| > \tau$, where $\tau > 0$ is the secret threshold value.

For response 0, squaring both sides,

$$
\begin{aligned}
&|\Delta_w - \Delta_r|^2 \leq t^2 \\
&\Rightarrow (\Delta_w - \Delta_r)^2 \leq t^2 \\
&= \left(\sum_{i=0}^{32} (u_i \cdot x_i + p) - \sum_{i=0}^{32} (v_i \cdot x_i + q) \right)^2 \leq t^2 \\
&= \left(\sum_{i=0}^{32} (u_i \cdot x_i - v_i \cdot x_i) + (p - q) \right)^2 \leq t^2 \\
&= \left(\sum_{i=0}^{32} (u_i - v_i) \cdot x_i + (p - q) \right)^2 \leq t^2 \\
&= \left(\sum_{i=0}^{32} \sum_{j=0}^{32} (u_i - v_i) \cdot (u_j - v_j) \cdot x_i \cdot x_j + 2 \sum_{i=0}^{32} (u_i - v_i) \cdot x_i \cdot (p - q) + (p - q)^2 \right) \leq t^2
\end{aligned}
$$

Now, taking t^2 to the left-hand side:

$$
\Rightarrow \sum_{i=0}^{32} \sum_{j=0}^{32} (u_i - v_i) \cdot (u_j - v_j) \cdot x_i \cdot x_j + 2 \sum_{i=0}^{32} (u_i - v_i) \cdot x_i \cdot (p - q) + (p - q)^2 - t^2 \leq 0
$$

$$
\sum_{i=0}^{32} (u_i - v_i)^2 \cdot x_i^2 + \sum_{i=0}^{32} \sum_{\substack{j=0 \\ j \neq i}}^{32} (u_i - v_i) \cdot (u_j - v_j) \cdot x_i \cdot x_j + 2 \sum_{i=0}^{32} (u_i - v_i) \cdot x_i \cdot (p - q) + (p - q)^2 - t^2 \leq 0
$$

since x_i can take only 2 values, +1 or -1, so x_i^2 is always positive, making it a constant term.
Thus, we can take

$$
\sum_{i=0}^{32} \sum_{\substack{j=0 \\ j \neq i}}^{32} (u_i - v_i) \cdot (u_j - v_j) \cdot x_i \cdot x_j + 2 \sum_{i=0}^{32} (u_i - v_i) \cdot x_i \cdot (p - q) + (p - q)^2 + \sum_{i=0}^{32} (u_i - v_i)^2 \cdot x_i^2 - t^2 \leq 0
$$

Therefore, the above equation for response 0 can be represented as,

$$
\mathbf{W}^\top \phi(\mathbf{c}) + b \leq 0
$$

where,

$$
\phi(\mathbf{c}) = [x_1 x_2, x_1 x_3, \ldots, x_1 x_{32}, x_2 x_3, \ldots, x_2 x_{32}, \ldots, x_{31} x_{32}, x_1, x_2, \ldots, x_{32}],
$$

with $x_i = (1 - 2c_i) \cdot (1 - 2c_{i+1}) \cdot \ldots \cdot (1 - 2c_{32}) \ \forall \ i \in [0, 32]$, and b is the constant term given by

$$
b = (p - q)^2 + \sum_{i=0}^{32} (u_i - v_i)^2 \cdot (1 - 2c_i)^2 \cdot (1 - 2c_{i+1})^2 \cdot \ldots \cdot (1 - 2c_{32})^2 \ - t^2
$$

Similarly, for response 1,

$$
\mathbf{W}^\top \phi(\mathbf{c}) + b > 0
$$

Thus, for any CAR-PUF, there exists a D-dimensional linear model $\mathbf{W} \in \mathbb{R}^D$ (D is 528 in our case) and a bias term $b \in \mathbb{R}$ such that for all CRPs $(\mathbf{c}, r)$ with $\mathbf{c} \in \{0, 1\}^{32}, r \in \{0, 1\}$, we have,

$$
\frac{1 + \text{sign}(\mathbf{W}^\top \phi(\mathbf{c}) + b)}{2} = r \ ; \ r \text{ is the response}
$$

2 Hyper-Parameter Tuning

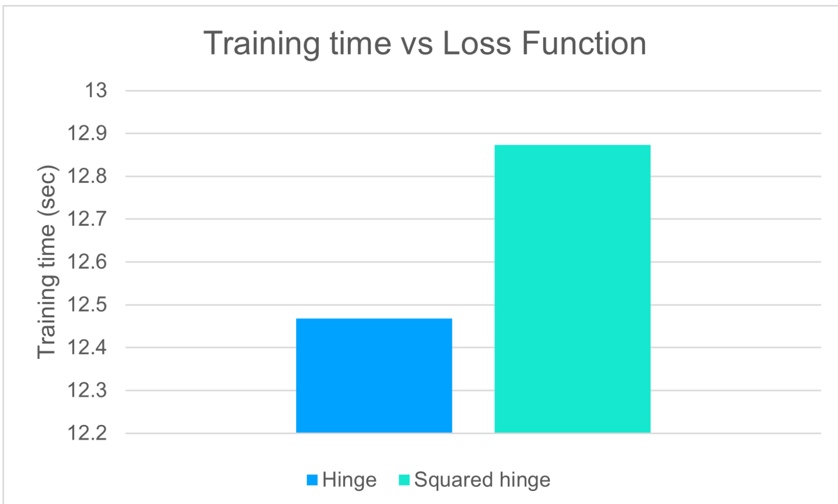
"Selecting an optimal value for a hyper-parameter is often considered an art rather than a science."

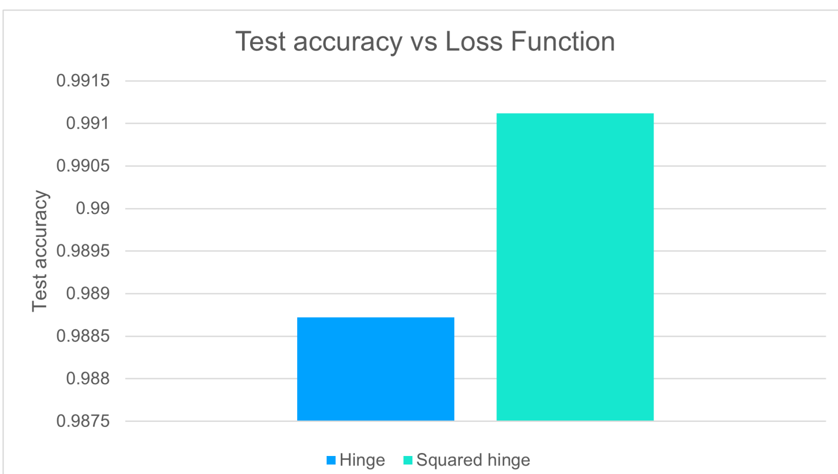
Hence, there exists no fixed algorithm to find out the best hyper-parameters. Instead, it depends on its utility in the Machine Learning model .

The hyper-parameters used in our attempt are as follows:

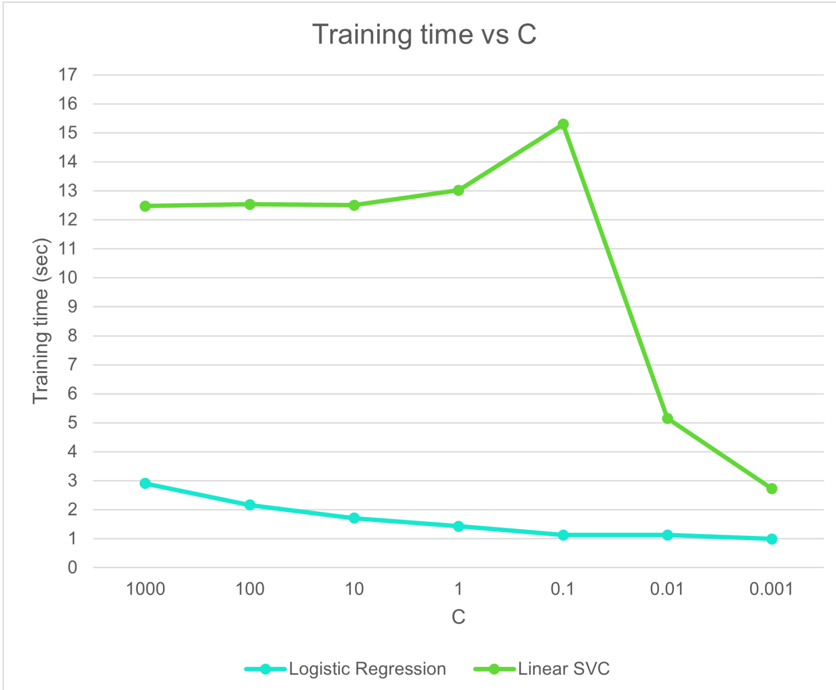
1. Loss (hinge vs squared hinge)
2. C
3. Tolerance (tol)

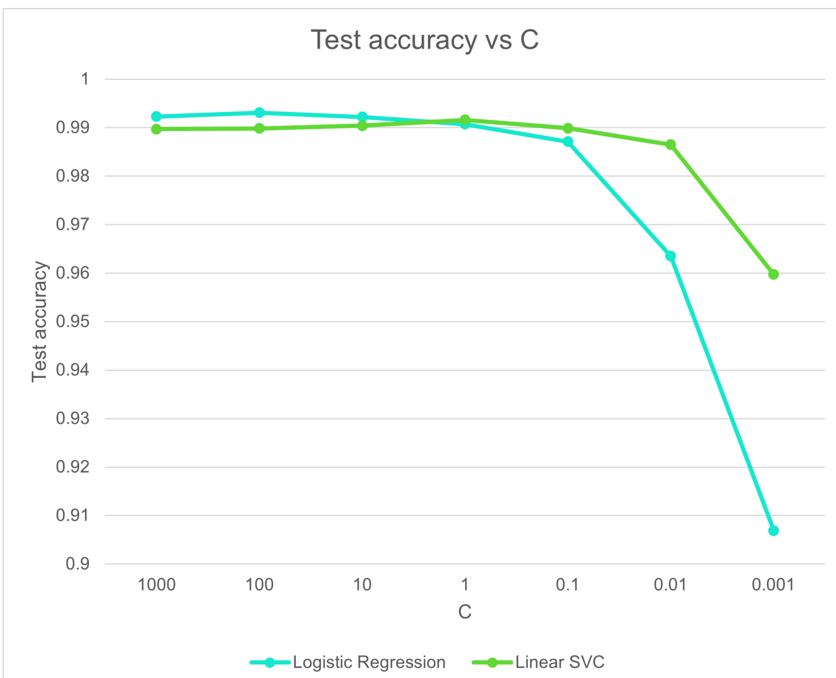
2.1 Loss (hinge vs squared hinge)



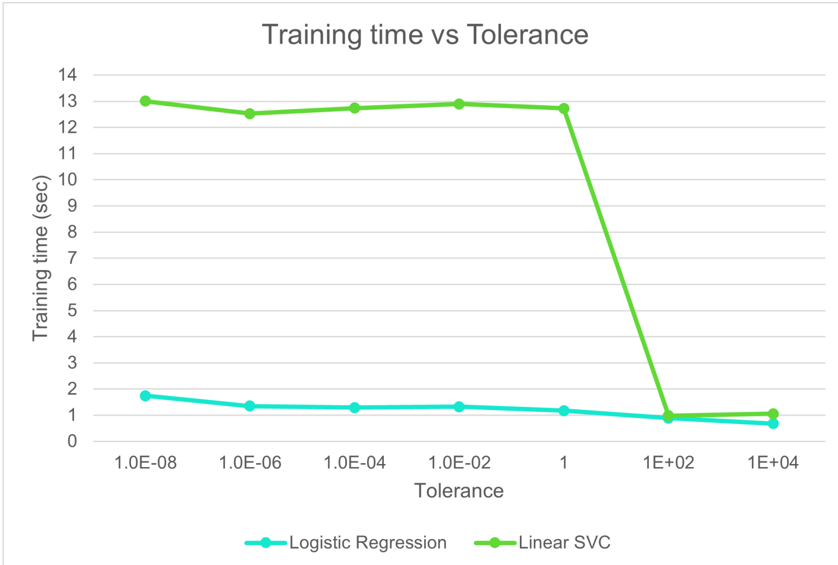


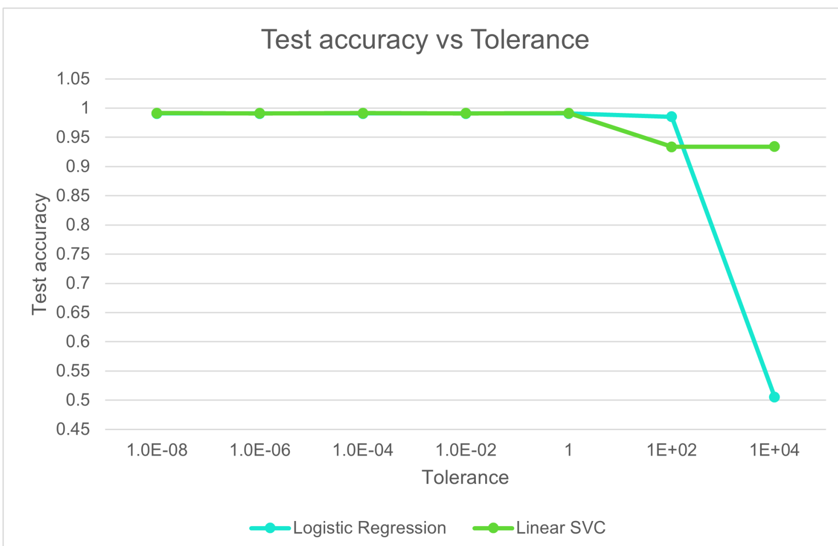
2.2 C


Training time vs C
Training time (sec)
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1000
100
10
1
0.1
0.01
0.001
C
Logistic Regression
Linear SVC


Test accuracy vs C
Test accuracy
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0.9
1000
100
10
1
0.1
0.01
0.001
C
Logistic Regression
Linear SVC

2.3 Tolerance (tol)





3 Conclusion

- It can be concluded that there exists a linear model that can predict the responses of a Companion Arbiter PUF with a very high accuracy, even though the delay difference depends on a secret value.
- It can be noticed that Logistic Regression requires less time to converge as compared to LinearSVC.
- The time taken to train the model decreases with increasing tolerance, which should be the case.
- Squared Hinge Loss Function gives more accuracy while taking more time.

References

[1] NumPy User Guide

[2] LinearSVC manual

[3] LinearRegression manual