CS771A Assignment I [CS771A_Group7]

Aryan Yadav
Department of Economics
210207
aryany21@iitk.ac.in

Priyanshu Kumar
Department of Material Science & Engineering
210783
priyanshuk21@iitk.ac.in

Saksham Gupta
Department of Electrical Engineering
210907
sakshamg21@iitk.ac.in

Shivam Gupta
Department of Mechanical Engineering
210982
shgupta21@iitk.ac.in

Umesh Tayde
Department of Mechanical Engineering
211126
umesht21@iitk.ac.in

Abstract

The Cross Connection Physically Unclonable Functions (**COCO-PUF**) is an innovative cryptographic primitive extending an Arbiter PUF's complexity. It consists of a chain of switch PUFs. In an Arbiter PUF, each switch PUF introduces delays that are difficult to replicate, and it selectively swaps signals based on challenger bits to generate a unique response to challenges. The COCO-PUF consists of two such PUFs, where **the time delay response of the upper signal** of the working Arbiter PUF is compared with that of a reference PUF and produces a binary response based on a **signal that advances**. This document presents a detailed analysis of the mathematical formulation of COCO-PUF, showing that it can be broken into **single linear models** to predict the responses of a COCO-PUF and such linear models can be estimated fairly accurately if given enough challenge-response pairs (CRPs).

QUESTION 1 & QUESTION 2

Arbiter PUF

An Arbiter PUF is a chain of k multiplexers, each of which either swaps the lines or keeps them intact, depending on the challenge bit fed into that multiplexer. The multiplexers each have delays which are hard to replicate but consistent. If the challenger bit is 0, the switch simply lets the signal pass through, but if the select bit is 1, the switch swaps the two signal paths. The switch PUF each has delays that are hard to replicate but consistent. Let t^u, t^l respectively denote the time for the upper and lower signals to reach the finish line. At the finish line resides an arbiter (usually a flip-flop) deciding which signal has reached first, the upper or the lower signal. The arbiter then uses this decision to generate the response.

C_i is the i-th challenger bit, where $C_i \in \{0, 1\}$
p_i = delay introduced by the i-th mux when the upper signal passes from above
q_i = delay introduced by the i-th mux when the lower signal passes from below
r_i = delay introduced by the i-th mux when the upper signal passes from below
s_i = delay introduced by the i-th mux when the lower signal passes from above

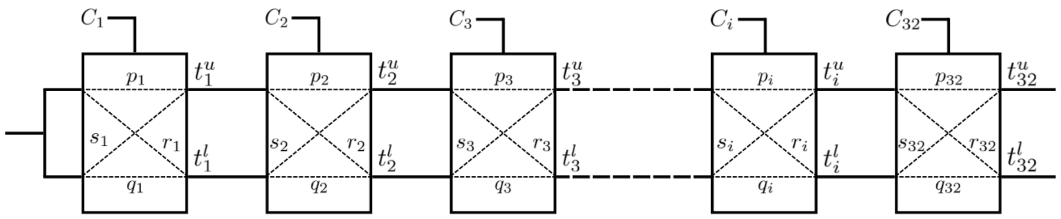


Figure 1: A simple arbiter PUF with 32 multiplexers

The time taken by the upper and lower signal to pass i-th mux depends only on the time taken to pass the previous mux, the delay introduced in the i-th mux and the challenger bit fed in the i-th mux as:

$$t_i^u = (1 - C_i) \cdot (t_{i-1}^u + p_i) + C_i \cdot (t_{i-1}^l + s_i)$$
$$t_i^l = (1 - C_i) \cdot (t_{i-1}^l + q_i) + C_i \cdot (t_{i-1}^u + r_i)$$

where,
t_i^u = time at which the upper signal leaves the i-th mux
t_i^l = time at which the lower signal leaves the i-th mux

Now, t_i^u can be rearranged as:

$$t_i^u = t_{i-1}^u - C_i \cdot (t_{i-1}^u - t_{i-1}^l) + p_i + C_i \cdot (s_i - p_i)$$
$$\implies t_i^u = t_{i-1}^u - C_i \cdot (\Delta_{i-1}) + p_i + C_i \cdot (\gamma_i)$$

where,
Δ_i = time lag in the signals at the i-th mux, i.e. $\Delta_i = t_i^u - t_i^l$
$\gamma_i = s_i - p_i$

Also, Δ_i can be rearranged as:

$$\Delta_i = t_i^u - t_i^l$$
$$\implies \Delta_i = (1 - C_i) \cdot (t_{i-1}^u + p_i - t_{i-1}^l - q_i) + C_i \cdot (t_{i-1}^l + s_i - t_{i-1}^u - r_i)$$
$$\implies \Delta_i = (1 - 2C_i) \cdot (t_{i-1}^u - t_{i-1}^l) + (1 - 2C_i) \cdot \left(\frac{p_i - q_i + r_i - s_i}{2}\right) + \left(\frac{p_i - q_i - r_i + s_i}{2}\right)$$
$$\implies \Delta_i = d_i \cdot \Delta_{i-1} + d_i \cdot \alpha_i + \beta_i$$

where,
$d_i = (1 - 2C_i)$
$\alpha_i = \left(\frac{p_i - q_i + r_i - s_i}{2}\right)$
$\beta_i = \left(\frac{p_i - q_i - r_i + s_i}{2}\right)$

Assuming t_0^u, t_0^l and Δ_0 to be 0. The time taken by the upper signal to reach finish line and the difference in the timings of the upper and the lower signal at the finish line can be expressed as follows:
For $i = 1$, we have,

$$t_1^u = t_0^u - C_1 \cdot (\Delta_0) + p_1 + C_1 \cdot (\gamma_1)$$
$$\implies t_1^u = p_1 + C_1 \cdot \gamma_1 \qquad \text{(Substituting } t_0^u \text{ and } \Delta_0 \text{ and simplifying)}$$
$$\Delta_1 = d_1 \cdot \Delta_0 + d_1 \cdot \alpha_1 + \beta_1$$
$$\implies \Delta_1 = d_1 \cdot \alpha_1 + \beta_1 \qquad \text{(Substituting } \Delta_0 \text{ and simplifying)}$$

For $i = 2$, we have,

$$t_2^u = t_1^u - C_2 \cdot (\Delta_1) + p_2 + C_2 \cdot (\gamma_2)$$

$$\implies t_2^u = [p_1 + C_1 \cdot \gamma_1] - C_2 \cdot [d_1 \cdot \alpha_1 + \beta_1] + p_2 + C_2 \cdot (\gamma_2) \quad \text{(Substituting } t_1^u \text{ and } \Delta_1\text{)}$$

$$\implies t_2^u = (p_1 + p_2) + C_1 \cdot \gamma_1 + C_2 \cdot (\gamma_2 - \beta_1) - (C_2 \cdot d_1) \cdot \alpha_1 \qquad \text{(Simplifying)}$$

$$\Delta_2 = d_2 \cdot \Delta_1 + d_2 \cdot \alpha_2 + \beta_2$$

$$\implies \Delta_2 = d_2 \cdot [d_1 \cdot \alpha_1 + \beta_1] + d_2 \cdot \alpha_2 + \beta_2 \qquad \text{(Substituting } \Delta_1\text{)}$$

$$\implies \Delta_2 = (d_2 \cdot d_1) \cdot \alpha_1 + d_2 \cdot (\alpha_2 + \beta_1) + \beta_2 \qquad \text{(Simplifying)}$$

For $i = 3$, we have,

$$t_3^u = t_2^u - C_3 \cdot (\Delta_2) + p_3 + C_3 \cdot (\gamma_3)$$

$$\begin{aligned}\implies t_3^u = {} & [(p_1 + p_2) + C_1 \cdot \gamma_1 + C_2 \cdot (\gamma_2 - \beta_1) - (C_2 \cdot d_1) \cdot \alpha_1] \\ & - C_3 \cdot [(d_2 \cdot d_1) \cdot \alpha_1 + d_2 \cdot (\alpha_2 + \beta_1) + \beta_2] \\ & + p_3 + C_3 \cdot (\gamma_3) \qquad \text{(Substituting } t_2^u \text{ and } \Delta_2\text{)}\end{aligned}$$

$$\begin{aligned}\implies t_3^u = {} & (p_1 + p_2 + p_3) + C_1 \cdot \gamma_1 + C_2 \cdot (\gamma_2 - \beta_1) + C_3 \cdot (\gamma_3 - \beta_2) \\ & - (C_2 \cdot d_1 + C_3 \cdot d_2 \cdot d_1) \cdot \alpha_1 - (C_3 \cdot d_2) \cdot (\alpha_2 + \beta_1) \qquad \text{(Simplifying)}\end{aligned}$$

$$\Delta_3 = d_3 \cdot \Delta_2 + d_3 \cdot \alpha_3 + \beta_3$$

$$\implies \Delta_3 = d_3 \cdot [(d_2 \cdot d_1) \cdot \alpha_1 + d_2 \cdot (\alpha_2 + \beta_1) + \beta_2] + d_3 \cdot \alpha_3 + \beta_3 \quad \text{(Substituting } \Delta_2\text{)}$$

$$\implies \Delta_3 = (d_3 \cdot d_2 \cdot d_1) \cdot \alpha_1 + d_3 \cdot d_2 \cdot (\alpha_2 + \beta_1) + d_3 \cdot (\alpha_3 + \beta_2) + \beta_3 \quad \text{(Simplifying)}$$

For $i = 4$, we have,

$$t_4^u = t_3^u - C_4 \cdot (\Delta_3) + p_4 + C_4 \cdot (\gamma_4)$$

$$\begin{aligned}\implies t_4^u = {} & [(p_1 + p_2 + p_3) + C_1 \cdot \gamma_1 + C_2 \cdot (\gamma_2 - \beta_1) + C_3 \cdot (\gamma_3 - \beta_2) \\ & - (C_2 \cdot d_1 + C_3 \cdot d_2 \cdot d_1) \cdot \alpha_1 - (C_3 \cdot d_2) \cdot (\alpha_2 + \beta_1)] \\ & - C_4 \cdot [(d_3 \cdot d_2 \cdot d_1) \cdot \alpha_1 + d_3 \cdot d_2 \cdot (\alpha_2 + \beta_1) + d_3 \cdot (\alpha_3 + \beta_2) + \beta_3] \\ & + p_4 + C_4 \cdot (\gamma_4) \qquad \text{(Substituting } t_3^u \text{ and } \Delta_3\text{)}\end{aligned}$$

$$\begin{aligned}\implies t_4^u = {} & (p_1 + p_2 + p_3 + p_4) + C_1 \cdot \gamma_1 + C_2 \cdot (\gamma_2 - \beta_1) + C_3 \cdot (\gamma_3 - \beta_2) \\ & + C_4 \cdot (\gamma_4 - \beta_3) - (C_2 \cdot d_1 + C_3 \cdot d_2 \cdot d_1 + C_4 \cdot d_3 \cdot d_2 \cdot d_1) \cdot \alpha_1 \\ & - (C_4 \cdot d_3 \cdot d_2 + C_3 \cdot d_2) \cdot (\alpha_2 + \beta_1) + (C_4 \cdot d_3) \cdot (\alpha_3 + \beta_2) \qquad \text{(Simplifying)}\end{aligned}$$

$$\begin{aligned}\implies t_4^u = {} & \left[\sum_{i=1}^{4} p_i\right] + \left[C_1 \cdot \gamma_1 + \sum_{i=2}^{4} C_i \cdot (\gamma_i - \beta_{i-1})\right] \\ & + \left[\left\{\sum_{j=1}^{3} C_{j+1} \cdot \left(\prod_{k=1}^{j} d_k\right)\right\} \cdot (-\alpha_1)\right. \\ & \left. + \sum_{i=2}^{3} \left\{\left\{\sum_{j=i}^{3} C_{j+1} \cdot \left(\prod_{k=i}^{j} d_k\right)\right\} \cdot (-\alpha_i - \beta_{i-1})\right\}\right]\end{aligned}$$

$$\implies t_4^u = b + W_1 \cdot \phi_1 + W_2 \cdot \phi_2 + W_3 \cdot \phi_3 + W_4 \cdot \phi_4 + W_5 \cdot \phi_5 + W_6 \cdot \phi_6 + W_7 \cdot \phi_7$$

where,

$$b = \sum_{i=1}^{4} p_i$$

$$W_i = \begin{cases} \gamma_i, & i = 1 \\ (\gamma_i - \beta_{i-1}), & 2 \le i \le 4 \\ -\alpha_{i-4}, & i = 5 \\ -(\alpha_{i-4} + \beta_{i-5}), & 6 \le i \le 7 \end{cases}$$

$$\phi_i = \begin{cases} C_i, & 1 \le i \le 4 \\ \left\{ \sum_{j=(i-4)}^{3} C_{j+1} \cdot \left(\prod_{k=(i-4)}^{j} d_k \right) \right\}, & 5 \le i \le 7 \end{cases}$$

In our case of 32-bit mux arbiter PUF, the time taken by the upper signal to reach the end of 32 multiplexers is given by:

$$\begin{aligned} t_{32}^u = & \left[\sum_{i=1}^{32} p_i \right] + \left[C_1 \cdot \gamma_1 + \sum_{i=2}^{32} C_i \cdot (\gamma_i - \beta_{i-1}) \right] \\ & + \left[\left\{ \sum_{j=1}^{31} C_{j+1} \cdot \left(\prod_{k=1}^{j} d_k \right) \right\} \cdot (-\alpha_1) \right. \\ & \left. + \sum_{i=2}^{31} \left\{ \left\{ \sum_{j=i}^{31} C_{j+1} \cdot \left(\prod_{k=i}^{j} d_k \right) \right\} \cdot (-\alpha_i - \beta_{i-1}) \right\} \right] \end{aligned}$$

$$t_{32}^u = b + W_1 \cdot \phi_1 + W_2 \cdot \phi_2 + W_3 \cdot \phi_3 + \cdots + W_{63} \cdot \phi_{63}$$

Thus, we can represent the time taken by the upper signal to reach the end of 32 multiplexers as,

$$t_{32}^u = \mathbf{W}^T \cdot \phi(\mathbf{c}) + b$$

where,

$$b = \sum_{i=1}^{32} p_i$$

$$W_i = \begin{cases} \gamma_i, & i = 1 \\ (\gamma_i - \beta_{i-1}), & 2 \le i \le 32 \\ -\alpha_{i-32}, & i = 33 \\ -(\alpha_{i-32} + \beta_{i-33}), & 34 \le i \le 63 \end{cases}$$

$$\phi(\mathbf{c})_i = \begin{cases} C_i, & 1 \le i \le 32 \\ \left\{ \sum_{j=(i-32)}^{31} C_{j+1} \cdot \left(\prod_{k=(i-32)}^{j} d_k \right) \right\}, & 33 \le i \le 63 \end{cases}$$

$d_k = (1 - 2C_k)$

From the above equation, we can infer that the feature map $\phi : \{0,1\}^{32} \to \mathbb{R}^D$, where $D = 63$. Thus, the dimensionality of $\mathbf{W}$ and ϕ is 63 and, total number of terms that are to be predicted equals $(63 + 1 \text{ (bias term)}) = 64$ terms.

Similarly, we can represent the time taken by the lower signal to reach the end of 32 multiplexers as,

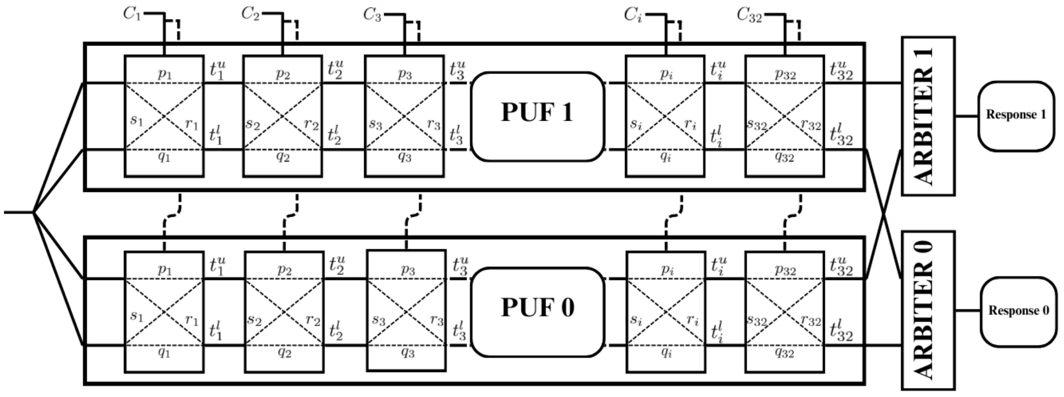
$$t_{32}^l = (\mathbf{W}^\mathbf{l})^T \cdot \phi(\mathbf{c}) + b^\mathrm{l}$$

And, the dimensionality of $\mathbf{W}_\mathrm{l}$ and ϕ is 63 and, total number of terms that are to be predicted equals $(63 + 1 \text{ (bias term)}) = 64$ terms.

QUESTION 3 & QUESTION 4

Cross-Connection PUF (COCO-PUF)

A COCO-PUF uses 2 arbiter PUFs, say PUF0 and PUF1 – each PUF has its own set of multiplexers with possibly different delays. When a challenge is provided, it is input into both PUFs. The response generation process differs from the conventional method. Instead of the lower and upper signals of PUF0 competing with each other, the lower signal from PUF0 competes with the lower signal from PUF1 using an arbiter, ARBITER0, to produce a response known as Response0. If the signal from PUF0 arrives first, Response0 is 0; if the signal from PUF1 arrives first, Response0 is 1. The upper signal from PUF0 is also made to compete with the upper signal from PUF1 using a second arbiter called ARBITER1 to generate a response called Response1. If the signal from PUF0 reaches first, Response1 is 0 else if the signal from PUF1 reaches first, Response1 is 1. Thus, on each challenge, the COCO-PUF generates two responses instead of one response.



C_1
C_2
C_3
C_i
C_{32}
PUF 1
PUF 0
ARBITER 1
ARBITER 0
Response 1
Response 0
p_1 s_1 r_1 q_1 t_1^u t_1^l
p_2 s_2 r_2 q_2 t_2^u t_2^l
p_3 s_3 r_3 q_3 t_3^u t_3^l
p_i s_i r_i q_i t_i^u t_i^l
p_{32} s_{32} r_{32} q_{32} t_{32}^u t_{32}^l

Figure 2: A COCO-PUF with 32-bit challenges and 2-bit responses

Let T_0^l, T_1^l be the time taken by the lower signal to reach the end of 32 multiplexers of PUF0 and PUF1 respectively, on the same challenge, i.e. same feature map and the corresponding response is Response0 ($r^0(\mathbf{c})$), which is 0 if $T_0^l - T_1^l < 0$ and it is 1 if $T_0^l - T_1^l > 0$.
Hence,

$$T_0^l - T_1^l < 0 \implies r^0(\mathbf{c}) = 0 \qquad\qquad T_0^l - T_1^l > 0 \implies r^0(\mathbf{c}) = 1$$

Thus, we need to figure out the sign of $(T_0^l - T_1^l)$, hence, the Response0 of the COCO-PUF can be expressed as,

$$r^0(\mathbf{c}) = \frac{1 + \text{sign}(T_0^l - T_1^l)}{2}$$

From QUESTION 1 & 2, the time taken by the lower signal to reach the end of 32 multiplexers as,

$$t_{32}^l = (\mathbf{W^l})^T \cdot \phi(\mathbf{c}) + b^l$$

Thus, the time taken by the lower signal to reach the end of 32 multiplexers of PUF0 can be represented as,

$$T_0^l = (\mathbf{W_0^l})^T \cdot \phi(\mathbf{c}) + b_0^l$$

And, the time taken by the lower signal to reach the end of 32 multiplexers of PUF1 can be represented as,

$$T_1^l = (\mathbf{W_1^l})^T \cdot \phi(\mathbf{c}) + b_1^l$$

As the challenge is same for both the PUFs, the feature map $\phi(\mathbf{c})$ remains the same for both PUFs. Hence,

$$T_0^l - T_1^l = \left[(\mathbf{W_0^l})^T \cdot \phi(\mathbf{c}) + b_0^l\right] - \left[(\mathbf{W_1^l})^T \cdot \phi(\mathbf{c}) + b_1^l\right]$$

$$\implies T_0^l - T_1^l = \left[(\mathbf{W_0^l})^T - (\mathbf{W_1^l})^T\right] \cdot \phi(\mathbf{c}) + \left[b_0^l - b_1^l\right]$$

$$\implies T_0^l - T_1^l = \left[(\mathbf{W_0^l}) - (\mathbf{W_1^l})\right]^T \cdot \phi(\mathbf{c}) + \left[b_0^l - b_1^l\right]$$

Thus, $(T_0^l - T_1^l)$ can be represented as,

$$(T_0^l - T_1^l) = \tilde{\mathbf{W}}^T \cdot \tilde{\phi}(\mathbf{c}) + \tilde{b}$$

where,
$\tilde{\mathbf{W}} = (\mathbf{W_0^l} - \mathbf{W_1^l})$ is the linear model
$\tilde{b} = (b_0^l - b_1^l)$ is the bias term
$\tilde{\phi}(\mathbf{c}) = \phi(\mathbf{c})$ is the feature map

Subsequently, the Response0 can be expressed as,

$$r^0(\mathbf{c}) = \frac{1 + \text{sign}(\tilde{\mathbf{W}}^T \cdot \tilde{\phi}(\mathbf{c}) + \tilde{b})}{2}$$

Similary, for Response1,

Let T_0^u, T_1^u be the time taken by the lower signal to reach the end of 32 multiplexers of PUF0 and PUF1 respectively, on the same challenge, i.e. same feature map and the corresponding response is Response1 ($r^1(\mathbf{c})$), which is 0 if $T_0^u - T_1^u < 0$ and it is 1 if $T_0^u - T_1^u > 0$.
Hence,

$$T_0^u - T_1^u < 0 \implies r^1(\mathbf{c}) = 0 \qquad T_0^u - T_1^u > 0 \implies r^1(\mathbf{c}) = 1$$

Thus, we need to figure out the sign of $(T_0^u - T_1^u)$, hence, the Response1 of the COCO-PUF can be expressed as,

$$r^1(\mathbf{c}) = \frac{1 + \text{sign}(T_0^u - T_1^u)}{2}$$

From QUESTION 1 & 2, the time taken by the lower signal to reach the end of 32 multiplexers as,

$$t_{32}^u = (\mathbf{W})^T \cdot \phi(\mathbf{c}) + b$$

Thus, the time taken by the lower signal to reach the end of 32 multiplexers of PUF0 can be represented as,

$$T_0^u = (\mathbf{W_0})^T \cdot \phi(\mathbf{c}) + b_0$$

And, the time taken by the lower signal to reach the end of 32 multiplexers of PUF1 can be represented as,

$$T_1^u = (\mathbf{W_1})^T \cdot \phi(\mathbf{c}) + b_1$$

As the challenge is same for both the PUFs, the feature map $\phi(\mathbf{c})$ remains the same for both PUFs.
Hence,

$$T_0^u - T_1^u = \left[(\mathbf{W_0})^T \cdot \phi(\mathbf{c}) + b_0\right] - \left[(\mathbf{W_1})^T \cdot \phi(\mathbf{c}) + b_1\right]$$

$$\implies T_0^u - T_1^u = \left[(\mathbf{W_0})^T - (\mathbf{W_1})^T\right] \cdot \phi(\mathbf{c}) + [b_0 - b_1]$$

$$\implies T_0^u - T_1^u = [(\mathbf{W_0}) - (\mathbf{W_1})]^T \cdot \phi(\mathbf{c}) + [b_0 - b_1]$$

Thus, $(T_0^u - T_1^u)$ can be represented as,

$$(T_0^u - T_1^u) = \tilde{\mathbf{W^u}}^T \cdot \tilde{\phi}(\mathbf{c}) + \tilde{b^u}$$

where,
$\tilde{\mathbf{W^u}} = (\mathbf{W_0} - \mathbf{W_1})$ is the linear model
$\tilde{b^u} = (b_0 - b_1)$ is the bias term
$\tilde{\phi}(\mathbf{c}) = \phi(\mathbf{c})$ is the feature map

Subsequently, the Response1 can be expressed as,

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

From the above equations, we can infer that a linear model can predict the Response0 and Response1 for a COCO-PUF. Also, the feature map $\tilde{\phi} : \{0,1\}^{32} \to \mathbb{R}^D$, where $D = 63$.
Thus, the dimensionality of $\tilde{\mathbf{W}}$ and $\tilde{\phi}$ is 63 and, total number of terms that are to be predicted equals $(63 + 1$ (bias term)$) = 64$ terms.

QUESTION 6

Loss hyperparameter in LinearSVC & corresponding training time and test accuracy

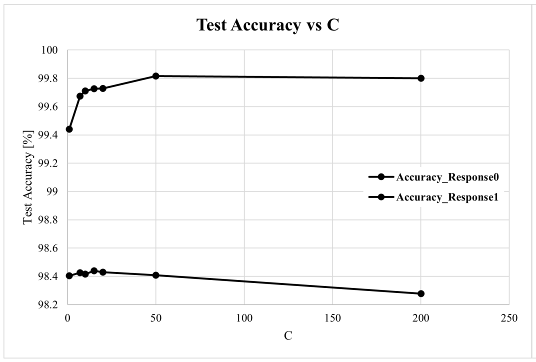
The data is tabulated below:

Loss Hyperparameter	**Training Time my_fit (in s)**	**Train Time my_map (in s)**	**Test Accuracy Response0 (in %)**	**Test Accuracy Response1 (in %)**
Hinge	2.706	0.121	98.404	99.440
Squared Hinge	19.263	0.066	98.000	99.790

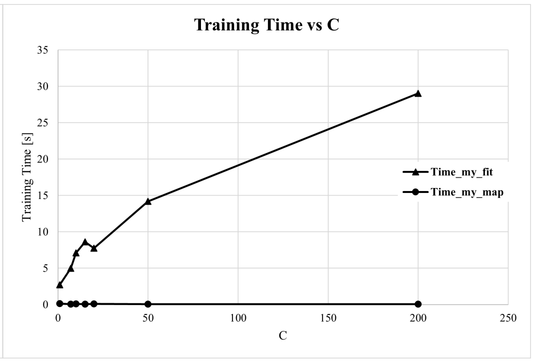
Table 1: Tabulated for C = 7, tolerance = 0.001, maximum iteration to be 10000

Training Time and Test Accuracy Dependence on C hyperparameter
Linear SVC

This data is set by fixing the following hyperparameters:
$loss =' hinge'$, $tol = 0.001$, and maximum iterations to be 10000.



(a) Test Accuracy vs. C

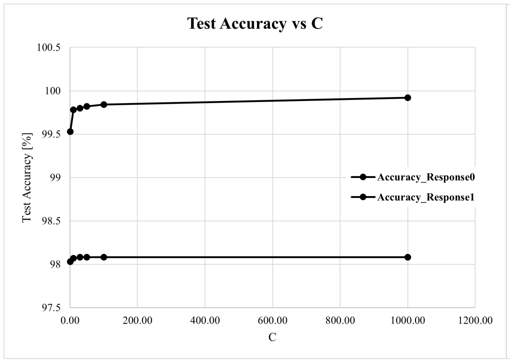


(b) Training Time vs. C

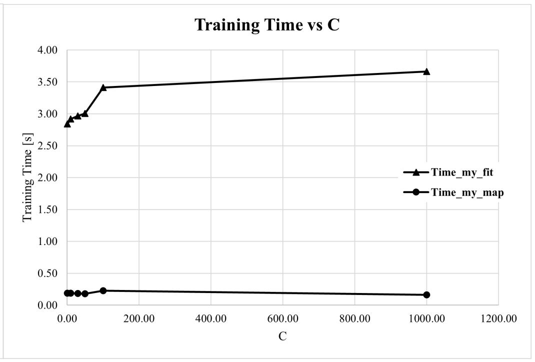
Figure 3: Comparison of Test Accuracy and Training Time vs. C

Logistic Regression

This data is set by fixing the following hyperparameters:
$tol = 0.001$, and maximum iterations to be 10000.



(a) Test Accuracy vs. C



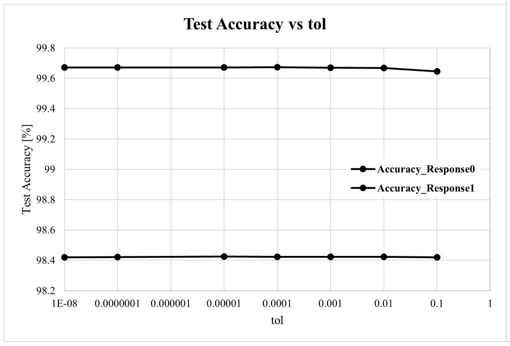
(b) Training Time vs. C

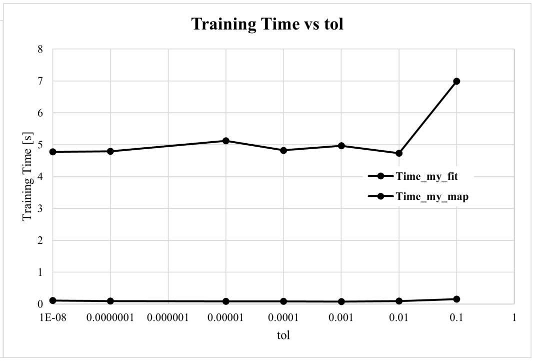
Figure 4: Comparison of Test Accuracy and Training Time vs. C

Training Time and Test Accuracy Dependence on *tol* hyperparameter

Linear SVC

This data is set by fixing the following hyperparameters:
$loss =' hinge'$, $C = 7$ and maximum iterations to be 10000.





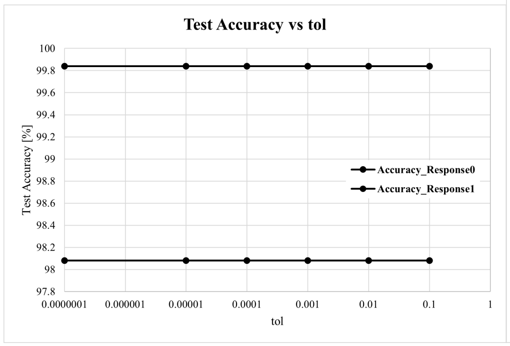
(a) Test Accuracy vs. C

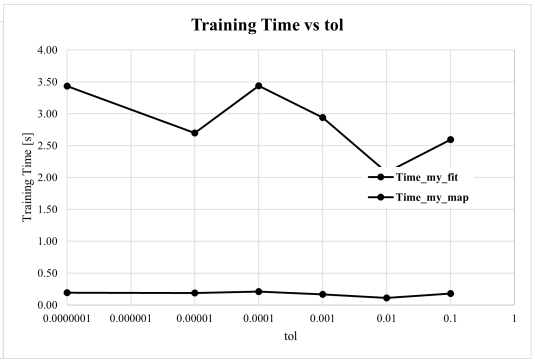
(b) Training Time vs. C

Figure 5: Comparison of Test Accuracy and Training Time vs. C

Logistic Regression

This data is set by fixing the following hyperparameters:
$C = 100$ and maximum iterations to be 10000.





(a) Test Accuracy vs. C

(b) Training Time vs. C

Figure 6: Comparison of Test Accuracy and Training Time vs. C