Problem 1

By giving a detailed mathematical derivation (as given in the lecture slides), show how a simple XORRO PUF can be broken by a single linear model. Recall that the simple XORRO PUF has just two XORROs and has no select bits and no multiplexers (see above figure and discussion on Simple XORRO PUF). Thus, the challenge to a simple XORRO PUF has just R bits. More specifically, give derivations for a map $\phi : \{0,1\}^R \to \mathbb{R}^D$ mapping R-bit 0/1-valued challenge vectors to D-dimensional feature vectors (for some D > 0) and show that for any simple XORRO PUF, there exists a linear model i.e. $\mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}$ such that for all challanges c $\in \{0,1\}^{\mathbb{R}}$, the following expression , the following expression

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

gives the correct response. (10 marks) .

Solution

For upper XORRO(XORRO): $f_0 = \frac{1}{{t_0}^0 + {t_1}^0}$
For upper XORRO(XORRO): $f_1 = \frac{1}{{t_0}^1 + {t_1}^1}$
For $i = 0, 1$, the time ${t_0}^i + {t_1}^i$ will be equal to sum of time delays caused by each XOR^j gate. The time delay caused by each XOR^j gate will be sum of $\delta^{i,j}_{0a_j} + \delta^{i,j}_{1a_j}$ as while the input received by XOR^j is 0 and signal bit is a_j it will cause delay by $\delta^{i,j}_{0a_j}$ and when the input is 1, the delay caused will be $\delta^{i,j}_{1a_j}$. As the input of the XORRO will oscillate between 0 and 1, the input of each XOR_j will also oscillate between 0 and 1 because each XOR_j will act either as identity or inverter.

$${t_0}^0 + {t_1}^0 = \sum_{j=0}^{R-1} (\delta^{i,j}_{0a_j} + \delta^{i,j}_{1a_j})$$

$$\implies \triangle = ({t_0}^1 + {t_1}^1) - ({t_0}^0 + {t_1}^0) = \frac{1}{f_1} - \frac{1}{f_0}$$

So, our answer will be:
$y = \frac{1+sign(\triangle)}{2}$
as if $f_0 > f_1, \triangle > 0$
and as if $f_1 > f_0, \triangle < 0$.
Now,

$$\triangle = ({t_0}^1 + {t_1}^0) - ({t_0}^0 + {t_1}^0)$$

$$= \sum_{j=0}^{R-1} (\delta^{1j}_{0a_j} + \delta^{1j}_{1a_j}) - \sum_{j=0}^{R-1} (\delta^{0j}_{0a_j} + \delta^{0j}_{1a_j})$$

$$= \sum_{j=0}^{R-1} (\delta^{1j}_{0a_j} + \delta^{1j}_{1a_j}) - \sum_{j=0}^{R-1} (\delta^{0j}_{0a_j} + \delta^{0j}_{1a_j})$$

$$= \sum_{j=0}^{R-1} (\delta^{1j}_{00} + \delta^{1j}_{10} - \delta^{0j}_{00} - \delta^{0j}_{10})(1 - a_j) + (\delta^{1j}_{11} + \delta^{1j}_{01} - \delta^{0j}_{11} - \delta^{0j}_{01})(a_j)$$

This is because when $a_j = 1$ delays would be δ^{ij}_{11} or δ^{ij}_{01}, depending on input and when $a_j = 0$, they would be δ^{ij}_{00} or δ^{ij}_{10}
So,

$$\triangle = \sum_{j=0}^{R-1} (w_j.a_j + b_j) = w^T.a + b$$

Where $w_j = \delta^{1j}_{11} - \delta^{0j}_{11} + \delta^{1j}_{01} - \delta^{0j}_{01} + \delta^{0j}_{00} - \delta^{1j}_{00} + \delta^{0j}_{10} - \delta^{1j}_{10}$
$b_j = \delta^{1j}_{00} - \delta^{0j}_{00} + \delta^{1j}_{10} - \delta^{1j}_{10}$

$a = (a_0, a_1, a_2, ..., a_{r-1})$
$w = (w_0, w_1, w_2, ..., w_{r-1})$
$b = (b_0, b_1, b_2, ..., b_{r-1})$

Finally substituting value of $\triangle = w^T.a + b$
We get y = $\frac{1+sign(\mathbf{w}^T.a+b)}{2}$
So, we get $\phi(a)$ = a or ϕ is an identity map from $(a_0, a_1, a_2, ..., a_{r-1})$ to $(a_0, a_1, a_2, ..., a_{r-1})$

Problem 2
Show how to extend the above linear model to crack an Advanced XORRO PUF. Do this by treating an advanced XORRO PUF as a collection of multiple simple XORRO PUFs. For example, you may use $M = 2^{S-1}(2^S - 1)$linear models, one for each pair of XORROs,to crack the advanced XORRO PUF. (10 marks)

Solution For this, we can create a dictionary of M linear models. We set the tuple (i,j) as the keys of the dictionary and map it to a linear model. Here $0 \leq j < S$ and $0 \leq i < j$ and hence the total number of models = $M = 2^{S-1}(2^S - 1)$.
Now, whenever MUX_0 chooses $XORRO_i$ as the first XORRO MUX_1 chooses $XORRO_j$ as the second XORRO where $i < j$, we can train model corresponding to key (i,j) with the Challenge-Bits (of length R) as input and counter response as target value. If, however j¡i, we can train model corresponding to key (j,i) with the Challenge-Bits (of length R) as input and (1 - counter response) as target value.
Again, while predicting the output, whenever MUX_0 chooses $XORRO_i$ as the first XORRO MUX_1 chooses $XORRO_j$ as the second XORRO where $i < j$, we can predict using model corresponding to key (i,j) and set predicted counter response as output. If, however j¡i, we can predict using model corresponding to key (j,i) and set predicted counter response as (1-output).

Problem 4
Report outcomes of experiments with both the svm.LinearSVC and LogisticRegression methods from sklearn library when used to learn the ensemble linear model. In particular, report how various hyper-parameters affected training time and test accuracy using tables, charts. Report these experiments with both LinearSVC and LogisticRegression methods even if your own submission uses just one of these methods or some totally different linear model learning method e.g. RidgeClassifier) In particular, you must report the affect of training time and test accuracy of at least 2 of the following:

a) changing the loss hyperparameter in LinearSVC (hinge vs squared hinge)

b) setting C hyperparameter in LinearSVC and LogisticRegression to high/low/medium values

c) changing the tol hyperparameter in LinearSVC and LogisticRegression to high/low/medium values

d) changing the penalty (regularization) hyperparameter in LinearSVC and LogisticRegression (l2 vs l1)

You may of course perform and report all the above experiments and/or additional experiments not mentioned above (e.g. changing the solver, maxiter etc) but reporting at least 2 of the above experiments

is required. You do not need to submit code for these experiments – just report your findings in the PDF file. Your submitted code should only include your final method (e.g. learning M models using LinearSVC) with hyperparameter settings that you found to work the best. (5 marks)

Solution For SVC :comparison between Square hinge loss(SH) and hinge loss(H)

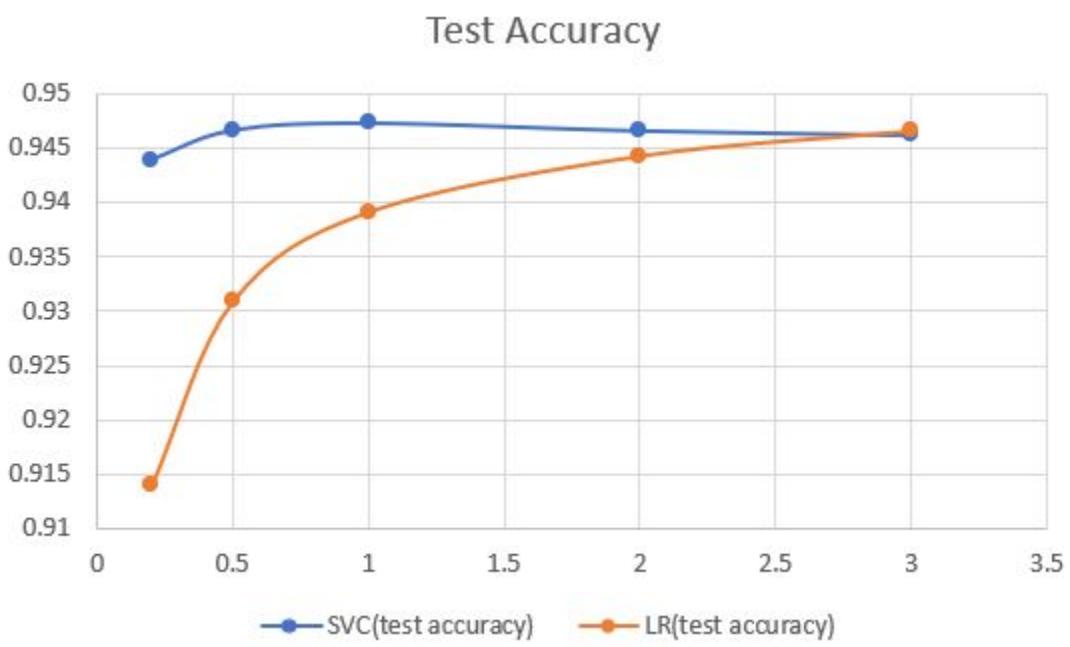
	Training time	Test Accuracy
SH	1.815607309	0.94735
H	1.41205883	0.939625

Comparing l2 vs l1 in both SVC and logistic regression

SVC	Training time	Test Accuracy
l1,dual=False	7.355065823	0.945975
l2,dual=False	0.913573742	0.947425

LR	Training time	Test Accuracy
l1,libniear	1.080396414	0.9353
l2,libniear	0.881072283	0.940375

C	SVC(training time	SVC(test accuracy)	LR(trainign time)	LR(test accuracy)
0.2	1.055137634	0.94395	1.334881067	0.914025
0.5	1.429440498	0.9467	1.49847436	0.930975
1	1.815607309	0.94735	1.629869699	0.939175
2	1.691559315	0.946625	1.638890982	0.944325
3	1.678859234	0.9462	1.791609287	0.946625


Test Accuracy
0.95
0.945
0.94
0.935
0.93
0.925
0.92
0.915
0.91
0
0.5
1
1.5
2
2.5
3
3.5
SVC(test accuracy)
LR(test accuracy)