Nithin S 221IT085

Remove Redundant Features and Reduce Dimensionality using Autoencoders and Train a SVM Model on different kernels.

Drop Repeated Columns

```
Column Index: 14, Column Name: tilde_count
Column Index: 28, Column Name: https_token
Column Index: 60, Column Name: brand_in_subdomain
Column Index: 86, Column Name: whois_registration
Column Index: 87, Column Name: domain_registration_length
Column Index: 88, Column Name: domain_age
Column Index: 96, Column Name: server_or_client_in_domain
Column Index: 98, Column Name: asn
Column Index: 99, Column Name: domain_activation_time
Column Index: 100, Column Name: domain_expiration_time
```

```
connection errors ratio,
'internal link tags ratio',
'sfh'.
'Nameservers count',
'Pop up windows',
'internal redirection ratio',
'External favicon',
'Internal media ratio',
'External errors ratio',
'External_redirection_count',
'dns record check',
'right click blocking',
'External redirection ratio',
'internal_errors_ratio',
'Domain in copyright',
'Average word length hostname',
'number of parameters',
'Vowel_count_in_domain',
'unsafe anchors',
'Media links ratio',
'login forms presence',
'Empty title',
'Invisible iframe',
'Submit to email',
'longest_word_hostname',
'external media ratio'
```

After this 98 features remain

Autoencoders to reduce Dimensionality to 15

Simple 2 layered architecture

```
Model: "model_28"

Layer (type) Output Shape Param #

input_15 (InputLayer) [(None, 97)] 0

dense_28 (Dense) (None, 15) 1470

dense_29 (Dense) (None, 97) 1552

Total params: 3,022
Trainable params: 3,022
Non-trainable params: 0
```

We have used a simple architecture for now due to the time it takes to train. Depending on how it performs we will increase the depth or keep it as it is.

Trained till 128 epochs with validation loss of 0.0002

```
Epoch 122: Training Loss = 0.0001, Validation Loss = 0.0002

Epoch 123: Training Loss = 0.0001, Validation Loss = 0.0002

Epoch 124: Training Loss = 0.0001, Validation Loss = 0.0002

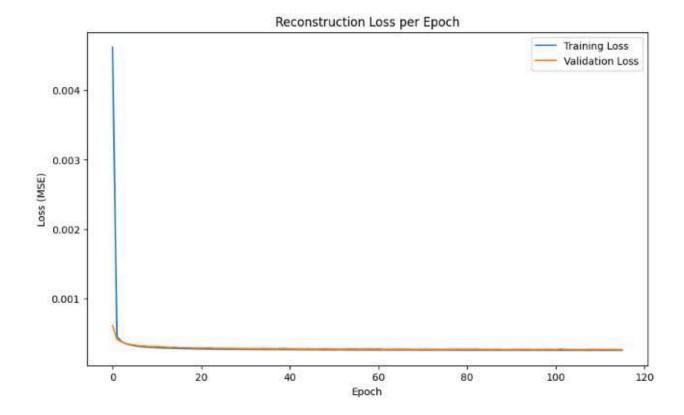
Epoch 125: Training Loss = 0.0001, Validation Loss = 0.0002

Epoch 126: Training Loss = 0.0001, Validation Loss = 0.0002

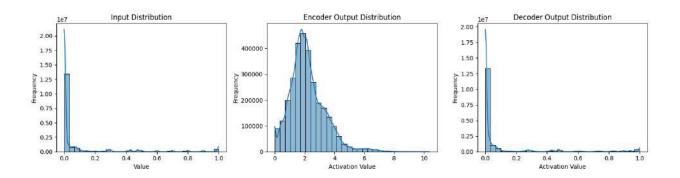
Epoch 127: Training Loss = 0.0001, Validation Loss = 0.0002

Epoch 128: Training Loss = 0.0001, Validation Loss = 0.0002
```

Final Data has 15 features extracted



Plots of input data to each layer



Weights of Extracted Features for the first 100 samples

Instead of directly mentioning numerical weights , I have color coded them for better understanding.

| | | | | | | Heatmap | | ded Feature | | | | | | | |
|-----------------|---|---|--|--|---|---|--|---|--|---|--|--|---|--|---|
| 0 - | 6.4 7.8 0 | 8.3 5.6 | 5.8 4.1 19 16 | 5.4 5 11 1.6 | 4.2 3.6 18 | 4.8 7.9 | 3.9 4.1 4.6 11 3.6 | 1,3 13 38 | 6.2 19 25 27 17 | 3.3 2.9 6.5 | 12 11 7.8 14 9.8 | 6.1 2 25 3.8 | 3.1 4.3 30 | 1.2 3 | 10 14 19 |
| 7 | 0 | 8.3 5.6 15 19 3.3 7.3 5.5 12 4.9 6.3 4.8 7.6 13 5.9 | 19 16 | 11 1.6 | 30 | 7.9 21 40 | 4.6 11 | 38 10 12 | 25 27 | 6.5 0 | 7.8 14 | 25 3.8 | 31 | 1.2 3 23 13 3.6 2.1 11 | 19 5.3 |
| 4 - | 7.2 | 3.3 | 4.5 | 5 5.9 | 2.6 | 7.5 5.6 | 3.6 | 12 0.65 | 17 | 3.8 3.6 | 9.8 12 | 0.72 | 1.2 | 3.6 2.1 | 13 |
| 9 - | 6.1 | 5.5 | 7.2 14 2.9 | 6.4 10 | 2.6 1.9 9.7 4.1 | 7.5 5.6 11 5.2 | 3.9 4.6 19 | 2.5 5.8 | 5.9 18 10 | 11 1.5 | 12 5 3 | 14 | 1.2 2.3 9 9 | 11 | 8.4 15 |
| co - | 7.2 7.2 6.1 4.2 7.5 5.6 | 4.9 | 4.3 25 6 | 4.8 | 4.4 | 8 | 4.4 | 14 | 18 | 3.7 37 | 12 | 6.5 14 1.2 1.3 17 4.6 | 4 20 | 3.8 | 13 |
| 10 | 6 | 4.8 | 6 | 5.5 | 24 0.54 | 20 5.6 | 2.7 | 12 14 | 18 | 2.8 | 12 | 4.6 | 0 | 2.3 | 16 |
| 17 | 5 | 13 | 9.6 10 6.5 | 3.2 | 13 | 6.6 8 8.1 | 4.4 3.4 2.7 2.5 0 4.2 | 6 10 | 18 29 18 21 16 5.3 | 7.5 | 13 | 3.1 11 9 | 4.2 14 3.9 | 5.4 | 14 |
| 4 - | 4.4 5 5.6 1.8 2.9 6.1 6.5 | 6.6 | 86 | 4.8 13 5.5 5.1 3.2 6.2 4.2 0 | 4.9 13 2.7 12 13 10 4.6 6.8 8.4 6.4 | 9.7 | 5.6 | 15 | 26 | 2.8 2.9 7.5 6.7 11 6.5 5.9 4.7 7.3 5.7 7.3 5.7 2.5 0 5.8 0.8 | 127 5.3 0 122 123 6.9 146 11.1 9.1 146 9.4 11.1 11.1 11.1 9.4 11.1 11.1 11.1 1 | 6.8 | 14 | 3.8 10 2.3 4.2 5.4 2.2 5.5 15 11 4.5 3.5 8.6 | 14 |
| 16 | 6.1 | 6.6 10 2.9 3.8 6.1 18 9.6 12 | 4.8 7.8 4 | 5.3 | 10 | 9.7 34 10 | 3 | 12 | 26 24 23 19 15 | 5.4 | 11 | 6.8 9.7 5.1 1.8 3.6 | 14 15 5.3 5.8 8.3 3.7 8 | 11 | 10 |
| 18 | 6.5 7 | 3.8 6.1 | 3.2 | 5.3 4.5 4.4 | 4.6 6.8 | 8.7 10 | 5.1 4.2 | 13 23 | 19 15 | 5.9 4.7 | 9.1 | 1.8 3.6 | 5.8 8.3 | 4.5 3.5 | 13 9.1 |
| 20 | 0 1 0 | 18 9.6 | 3.2 15 12 | 8.9 3.5 | 8.4 6.4 | 2.8 4.4 | 0 4.2 | 12 11 | 9.8 7.7 0 | 7.3 5.7 | 7 14 | 0.88 15 11 | 3.7 8 | 8.3 6.6 | 0 8.9 |
| 22 | 5.7 5.6 | 12 6.7 | 8.5 6.2 | 5.5 5.2 | 18 10 | 9.6 9.4 | 13 5.5 | 28 16 | 21 9.8 | 7.3 5.7 | 16 9.4 | 3.5 | 35 14 | 26 7.9 | 12 16 |
| 24 | 0 | 9.3 | 6.2 11 5.7 5.2 | 8.9 3.5 5.5 5.2 4.6 0 | 18 10 3.9 19 5.3 7.8 9.5 9.5 | 5.3 46 | 5.6 11 3 5.1 4.2 0 4.2 13 5.5 2.2 11 4.6 4.2 2.8 7.8 4.6 | 15 17 12 13 23 12 11 28 16 1.3 12 15 13 | 9.8 35 24 | 2.5 0 | 10 11 | 4.7 | 35 14 5.4 5.7 8.5 | 7.9 2.2 25 2.6 3.6 7.2 17 5 | 10 |
| - - | 5.6 7.5 8.3 11 | 8.6 6.2 | 5.2 4 | 6.3 3.3 3 6.4 6.7 | 5.3 7.8 | 8.8 | 4.6 4.2 | 15 13 | 24 3.1 | 5.2 6.3 | 13 11 | 9.5 3.5 | 8.5 9.5 | 2.6 3.6 | 16 3.5 |
| 28.3 | 8.3 11 | 13 7.2 | 5.1 11 10 | 3 6.4 | 9.5 9.5 | 8.5 19 | 2.8 7.8 | 3.4 16 19 | 3.1 13 27 31 | 5.8 0.8 | 9.9 13 | 11 5.8 | 9.5 8 17 20 | 7.2 17 | 8.4 9.1 |
| 30 2 | 5 9.4 | 12 7.4 | 10 5.3 | 6.7 5.2 | 16 2.7 | 5.8 7.1 | 4.6 4.3 | 19 0.63 | 31 5.6 | 8.4 3.2 | 15 9.1 | 10 3.8 | 20 4.4 | 5 1.4 | 16 9.1 |
| 32 | 5 9.4 4.4 7.5 0 | 5.2 6.4 | 5.3 9.3 5.1 0 | 3.6 4.3 | 2.5 3.9 | 5 7.6 | 2.1 3.6 | 12 14 | 5.6 20 20 32 | 3.5 2.7 | 12 10 | 3.5 4.7 22 9.5 3.5 11 5.8 10 3.8 2.2 1.5 | 1.7 4.5 | 2.3 3.1 | 14 13 |
| ¥ - | 0 8.4 | 23 6.4 | 6.3 | 5.2 3.6 4.3 11 6.3 4.8 3.1 6.1 | 2.7 2.5 3.9 13 4.6 2.6 | 9.5 8.5 19 5.8 7.1 5 7.6 17 11 8 8 | 57 5.9 | 11 9.3 | 32 | 3.2 3.5 2.7 8.5 5.4 5.1 6.2 3.4 6.1 | 7.5 | 18 4.7 | 4.4 1.7 4.5 8.4 7.2 7.7 | 1.4 2.3 3.1 15 5.6 3.7 7.8 3 | 0 4.1 |
| 36 - | 6.2 4.5 | 6.3 12 | 6.3 11 10 | 4.8 3.1 | 2.6 10 | 8.8 | 3.3 5.3 | 22 5.9 | 1.1 18 15 19 | 5.1 6.2 | 8.7 13 | 4.7 7.1 10 5.2 | 7.7 14 | 3.7 7.8 | 11 16 |
| 38 | 5.9 | 5.4 4.3 | 6.2 8.6 | 6.1 | 7.9 | 6.4 | 2.9 5.2 | 13 | 19 0 | 3.4 6.1 | 12 | 5.2 11 | 0.76 8 | 3 5.8 | 16 |
| 40 - | 8.4 6.2 4.5 5.9 3.7 0 7 6.4 5.4 | 6.7 9.3 6.2 13 7.2 7.4 5.4 6.4 6.3 12 5.4 8.8 16 7.5 11 8.9 5.6 | 25 | 11 | 64 | 6.4 4.4 27 14 | 2.5 7.8 | 24 28 | 39 30 | 0 | 9.1 12 10 0 7.5 8.7 13 12 7.1 2.2 3.7 8.4 17 16 14 | 53 | 40 | 64 | 7.9 |
| 4 - | 6.4 5.4 | 7.5 | 6.9 12 18 14 8.1 | 4.8 6.8 | 2.7 2.7 19 | 9.2 5.1 21 1.4 9.7 | 5 | 11 | 8 21 | 11 7.2 10 8 5.5 5.6 | 8.4 17 | 15 13 11 6.5 7.3 8.1 | 29 6.4 22 31 13 6.8 | 11 5.7 6 | 7.8 |
| 4 - | 9.2 7.6 | 13 | 18 | 7 8 1 | 4.9 11 | 21 | 11 | 10 | 14 | 8 | 16 14 | 6.5 | 31 | 23 | 19 15 |
| · · · · | 6 8.8 | 5.6 | 8.1 | 5 | 2 | 9.7 6.8 | 4.3 3.6 57 5.9 3.3 5.3 5.3 2.9 5.2 7.8 5 5.9 11 2.2 4.7 2.9 3.89 | 22 16 | 17 | 5.6 6.2 | 10 11 | 8.1 | 6.8 | 6.6 4.7 8.2 | 11 |
| Index 48 4 | 6.6 7.3 | 5.9 | 4.8 11 7.9 10 | 3.9 | 8.6 0 | 6.8 7.1 2.8 7.7 | 2.9 | 13 | 21 | 3.5 | 12 | 4.9 6.8 | 4.5 | 8.2 2.9 1.7 9.4 | 15 |
| | 6.5 | 9.7 12 | 10 | 7.6 9.5 | 2.2 3.4 4.3 | 7.7 8.6 | 0.89 | 15 16 | 39 30 8 21 14 27 17 27 21 9 23 23 | 6.2 3.5 4.4 8.9 4.4 | 12 9.8 | 2.9 1.1 | 6.3 | Q. | 13 |
| Sample 52 50 | 3 6.5 0.91 2.9 1.6 | 9.8 5.9 5.6 9.7 12 11 8.9 13 6.1 5.1 7.5 6.2 | 7.9 16 | 6.5 11 3.5 4.8 6.8 7 8.1 5 3.7 3.9 6 9.5 9.5 | 24 33 | 44 15 | 3 11 6.5 19 | 0.63 112 114 111 9.3 222 5.9 122 228 111 9.8 100 17 22 24 28 111 15 16 13 15 16 34 9.5 13 | 37 22 | 0 10 | 11 12 11 12 9.8 11 19 7.4 | 12 4.9 6.8 2.9 1.1 14 14 | 10 4.5 2.4 6.3 6.1 16 22 | 19 17 11 1.4 | 5,3 13 10 10 11 11 12 12 14 14 17,2 14 18 10 13 19 10 11 10 11 11 11 11 11 11 11 11 11 11 |
| N N | 1.6 6.2 | 13 | 7.4 5 | 9.8 | 6.9 2.3 | 4.5 6.4 | 19 4.3 | 13 | 5.6 3.6 | 3.9 4.6 | 7.4 8.5 | 9.8 7.3 3.1 5.8 | 0.0 | 11 | 14 7.5 |
| 5 - | 7.1 | 5.1 7.5 | | 5 5 4.6 | 4.2 6.9 | 9 7.7 | 4.7 4.5 | 13 0.6 | 20 7.3 | 5.2 6.3 | 8.5 10 9.5 | 3.1 5.8 | 3.5 6.1 4.2 | 3.5 6.2 | 13 8.1 |
| 58 - | 2.7 | 6.2 | 7.7 | 9.8 7.4 | 9.7 | 13 | 5.2 11 | | | 9.1 0 | 7.5 | 8 | 14 19 10 | 0.73 | 5.5 17 |
| 2 09 | 7.1 7.2 2.7 1.4 0 7.4 8.3 | 10 | 4.7 7.7 9.4 5.6 3.9 | 0 | 23 | 43 50 7.1 | 4.7 4.5 5.2 11 12 4.5 4.8 | 23 13 | 33 | 8.8 3.5 | 9.5 7.5 17 5.3 12 12 | 22 29 0.43 | 10 3.6 | 18 29 3.2 4.8 | 5.7 |
| 553 | 8.3 | 3.9 4.8 | 5.3 | 3.8 5.2 7.3 6.8 4.6 4.9 3.3 5.4 | 9.7 22 23 3.2 2.5 8.9 | 9.1 | 4.8 | 21 19 23 13 13 12 | 23 37 33 17 18 11 17 12 8.5 17 2.2 8.4 | 8.8 3.5 3.5 7.6 7.6 5.8 3.7 5.6 5.9 3.8 | 12 | 0.43 2.2 3.3 7.9 7.6 6.6 1.7 3.8 2.4 4.8 0.49 | 4.9 | 4.8 | 13 |
| 64 62 | 8.8 | 19 12 11 10 2.5 4.9 8.1 | 5.3 15 7.5 9.8 | 6.8 | 16 | 2.4 4.7 8.1 7.6 7.8 9.6 7.6 11 7.5 | 5.7 | 8.6 3 | 17 | 7.6 5.8 | 7.7 13 13 11 8.6 6.7 | 7.9 | 2.9 19 12 | 11 4.8 3.2 5 2.8 3.7 3.7 5.1 4 | 9,1 9,1 8,4 13,21 8,9 9,4 11,8,8 11,1 8,8 12,7 14,7,1 20,8 13,1 14,7,1 14,1 10,6,7 |
| 9 99 | 8.4 | 10 | 4.3 4.7 6.7 | 4.6 4.9 | 7.8 | 7.6 | 3.2 | 2.6 11 | 8.5 | 3.7 5.6 | 11 8 6 | 6.6 | 6 0.87 | 5 | 8.4 13 |
| | 8.8 5.1 8.4 7.3 7.6 7.3 5.6 6.9 | 4.9 | 6.7 | 3.3 | 16 7.7 7.8 1.5 8.6 7.9 10 3.4 | 9.6 | 5.9 | 0 | 2.2 | 5.9 | 6.7 | 3.8 | 9.3 6.6 | 3.7 | 21 |
| 70 68 | 5.6 | 8.3 | 5.2 4.3 4.3 | 4.7 | 10 | 11 | 5.9 | 4.8 12 12 | 5.3 17 18 | 6.2 | 5.3 11 | 4.8 | 12 2.3 | 5.1 | 9.4 |
| 100 | 5.2 | 11 | 4.3 | 1.4 | 11 | 10 33 | 5.6 | 4.2 | 18 | 12 | 9.8 | 14 | 60 | 13 | 8.2 |
| 7 - | 3.8 | 4.9 | 7.1 | 4.7 4.7 1.4 18 4.4 5 6.4 | 9.2 | 10 9.3 | 5.7 3.4 3.2 3.4 5.9 5.2 5.9 4.1 5.6 13 3.4 3.1 | 22 | 19 20 | 6.2 4.2 12 17 11 5.6 | 12 | 9.2 3.5 19 | 12 5.6 | 5.2 3.1 19 | 8.8 |
| 5 74 | 5.2 1.8 3.8 7.3 3 7.8 5.7 | 8.3 3.3 11 9.5 4.9 5.1 14 4.3 5.6 | 22 7.1 4.1 4.2 4.2 6.5 11 | 6,4 | 9.2 4.2 22 2.9 2.4 | 44 | 13 | 4.2 7.9 22 13 27 13 9.9 13 | 20 28 17 3.8 | D | 8 12 10 12 11 7.3 10 | 19 | - 19 | 19 | 9.9 |
| 3 76 | 5.7 4.9 | 5.6 | 6.5 | 4.4 6.1 2.6 0 | 2.4 4.2 | 7.3 7 6.5 | 3.8 3 | 9.9 | 3. 8 23 | 3.4 5.1 6.2 2 5.8 | 7.3 | 1.1 7.3 8.5 | 2.7 2 7.6 1.8 4.8 | 2.8 2.5 6.9 | 7.1 |
| 78 | 6 6.9 3.8 | 6.3 0 3.6 15 6.4 12 6.2 | 10 | 0 | 5.2 3.8 | 6.7 8.4 | | 0 12 | 6.4 18 | 2 | ii ii | 4 | 1.8 | 0 | 9.8 |
| 2 80 | 3.8 6.6 | 15 | 5.4 3.2 | 4.3 7.7 | 22 4.1 | 45 | 6.4 4.7 12 4.9 4.2 4.3 | 19 0.98 | 35 4.2 | | 14 14 | 2.1 20 7.2 6.7 7.2 | 18 0.73 | 0 3.5 19 4.4 | 14 |
| 4 82 | 4.2 5.7 | 12 | 8 11 | 8.2 4.6 | 9.6 | 6.3 7.2 | 4.2 | 14 14 | 8.5 | 4.1 3.9 6.7 | 9.4 | 6.7 | 10 | 2.1 4.1 | 6.7 |
| 84 | 6.6 | | | 4 5.6 | 3.2 10 | 8 13 | 6.3 | 1.6 | 24 9.2 | ii | 14 8.7 | 6.3 | 10 13 | 6.2 | 13 |
| 3 86 | 2.4 | 13 | 15 | 8.9 | 12 | 13 | 6.7 | 36 | 16 | 6.8 | 7.3 | 13 | 25 | 17 | 21 |
| 88 (| 7.1 | 7.4 | 4.8 | 5.2 | 8.3 | 8.2 | 4.6 | 8.7 | 3.6 | 7.5 | 5.1 | 7.8 | 5.6 | 6.1 | 4.3 |
| 96 - | ii | 8.9 8.9 | 9,2 | 8.8 8.8 | 11 | 14 0 | 6.3 3.3 6.7 3.9 4.6 6.7 3.5 | 4,4 | 5.3 10 | 6.3 | 12 | 7.6 5.5 | 18 | 4.8 | 8.3 |
| 76 t | 7.6 | 4.7 | 4.7 | 4.7 | 3.7 | 6.3 | 4.1 | 0 | 4 | 5.9 | 8.4 | 2 | 1.9 | 1.6 | 8.1 |
| g - | 6 | 2.8 | 4.1 | 5.1 | 5.3 | 10 | 4.8 | 25 20 | 14 | 8.5 7.7 | 5.Z | 3.3 3.4 | 4.8 | 9.8 4.1 | 9.1 |
| 96 - | 6.6 0 2.4 3.7 7.1 7.7 11 0 7.6 4.9 6 5.6 6.8 7.7 | 8.7 13 9 7.4 13 8.9 8.7 4.7 6.3 2.8 9.3 7.6 6.7 | 7.2 19 15 8.5 4.8 6 9.2 0 4.7 4 4.1 4.6 11 3.8 3.9 | 5.6 11 8.9 5.2 6.6 8.8 7.2 4.7 5.1 9.9 3.6 4.8 | 10 49 12 7.7 8.3 11 14 3.7 13 5.3 5.5 1.1 7 | 13 22 13 8.7 8.2 14 0 26 6.3 14 10 9.4 5.5 7.7 | 4.1 4.7 4.8 1.6 3.1 4.9 | 1.6 21 36 24 8.7 13 4.4 35 0 25 20 23 3.1 15 | 9.2 28 16 19 3.6 3.3 10 27 4 31 14 16 10 21 | 11 0 6.8 3.9 7.5 8.1 6.3 3.4 5.9 8.5 7.7 8.7 3.6 5.2 | 8.7 0 7.3 10 5.1 0.02 12 0 8.4 15 6.2 4 12 13 8.8 | 6.3 37 13 5.5 7.6 5.5 2 3.3 3.4 3.3 5.2 5.2 | 13 36 25 9 5.6 18 11 21 1.9 16 4.8 6.9 4.7 6.3 | 6.2 17 4.4 6.1 21 4.8 16 1.6 9.8 4.1 9.1 1.4 5.9 | 8.3 11 |
| 88 - | 3.5 | | 3.8 3.9 | 3 4.8 | 24 | 45 | | | 21 36 | | | | | 23 | 13 6.5 21 11 4.3 14 8.3 0 8.1 20 9.1 8.3 11 14 14 |
| | ò | i | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| | | | | | | | E | ncoded Featu | ii e | | | | | | |

SVM Model trained on those 15 extracted features using 4 kernels

Data Splitting First: The train-test split is performed using the original features, ensuring the autoencoder is not influenced by the test data.

Training Autoencoder on Training Data Only: This prevents any leakage from the test set into the autoencoder, leading to a more realistic evaluation.

Consistent Transformation: Both the training and test sets are transformed using the same trained encoder, ensuring consistent feature representation for the SVM.

```
--- Training SVM with kernel: linear ---
Accuracy: 0.9364
Confusion Matrix:
[[9843 560]
 [ 712 8885]]
Classification Report:
               precision
                            recall f1-score
                                                support
           Θ
                   0.93
                             0.95
                                       0.94
                                                10403
                   0.94
                             0.93
                                       0.93
                                                 9597
                                       0.94
                                                20000
    accuracy
                   0.94
                             0.94
                                       0.94
                                                20000
   macro avg
                   0.94
weighted avg
                             0.94
                                       0.94
                                                20000
```

```
--- Training SVM with kernel: poly ---
Accuracy: 0.963
Confusion Matrix:
[[10041 362]
[ 378 9219]]
Classification Report:
                         recall f1-score
              precision
                                             support
          Θ
                  0.96
                           0.97
                                     0.96
                                              10403
          1
                  0.96
                            0.96
                                     0.96
                                               9597
                                     0.96
   accuracy
                                              20000
  macro avg
                                     0.96
                  0.96
                            0.96
                                              20000
weighted avg
                  0.96
                            0.96
                                     0.96
                                              20000
```

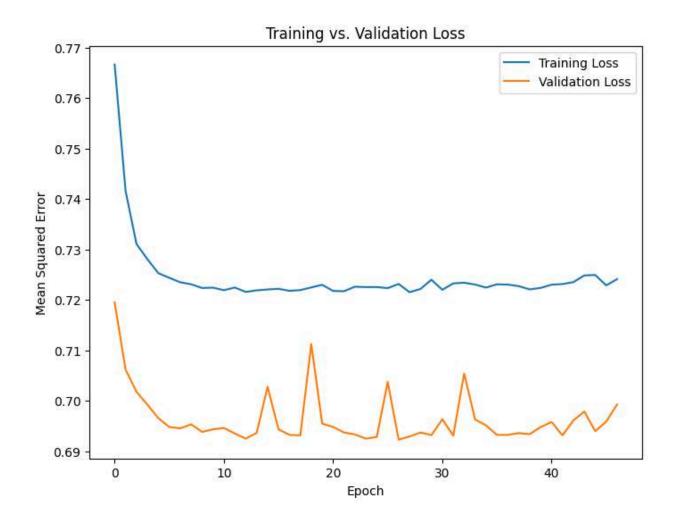
| Accuracy: 6 | .972 | 225 | | | |
|-------------|-------|-----------|--------|----------|---------|
| Confusion M | latri | ix: | | | |
| [[10199 | 204 | Ì | | | N |
| [351 92 | 46]] | | | | W2 |
| Classificat | ion | Report: | | | |
| | | precision | recall | fl-score | support |
| | Θ | 0.97 | 0.98 | 0.97 | 10403 |
| | 1 | 0.98 | 0.96 | 0.97 | 9597 |
| accurac | y | | | 0.97 | 20000 |
| macro av | /g | 0.97 | 0.97 | 0.97 | 20000 |
| weighted av | g | 0.97 | 0.97 | 0.97 | 20000 |

```
--- Training SVM with kernel: sigmoid ---
Accuracy: 0.3808
Confusion Matrix:
[[4199 6204]
[6180 3417]]
Classification Report:
              precision recall f1-score
                                             support
          0
                           0.40
                                     0.40
                  0.40
                                              10403
          1
                  0.36
                            0.36
                                     0.36
                                               9597
   accuracy
                                     0.38
                                              20000
                  0.38
                            0.38
                                     0.38
  macro avg
                                              20000
weighted avg
                  0.38
                            0.38
                                     0.38
                                              20000
```

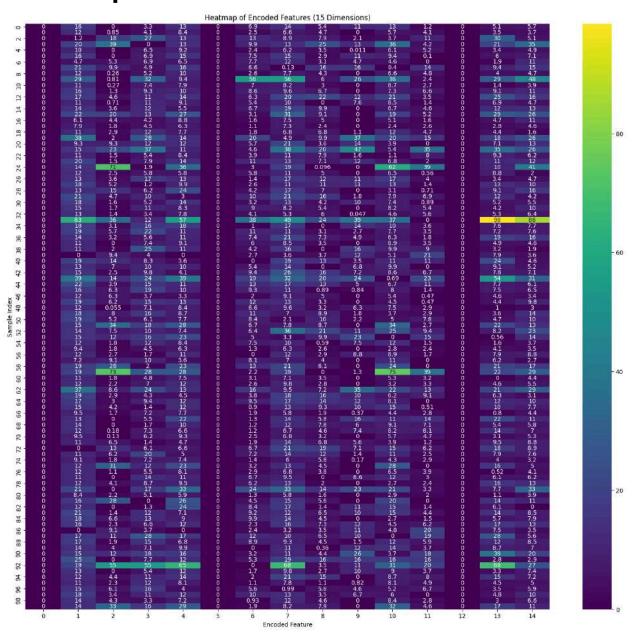
Complex Autoencoder architecture

| Layer (type) | Output Shape | Param # |
|---------------------------|--------------|---------|
| input_layer (InputLayer) | (None, 97) | 0 |
| encoder_dense_128 (Dense) | (None, 128) | 12,544 |
| encoder_dense_64 (Dense) | (None, 64) | 8,256 |
| encoder_dense_32 (Dense) | (None, 32) | 2,080 |
| encoder_output (Dense) | (None, 15) | 495 |
| decoder_dense_32 (Dense) | (None, 32) | 512 |
| decoder_dense_64 (Dense) | (None, 64) | 2,112 |
| decoder_dense_128 (Dense) | (None, 128) | 8,320 |
| decoder output (Dense) | (None, 97) | 12,513 |

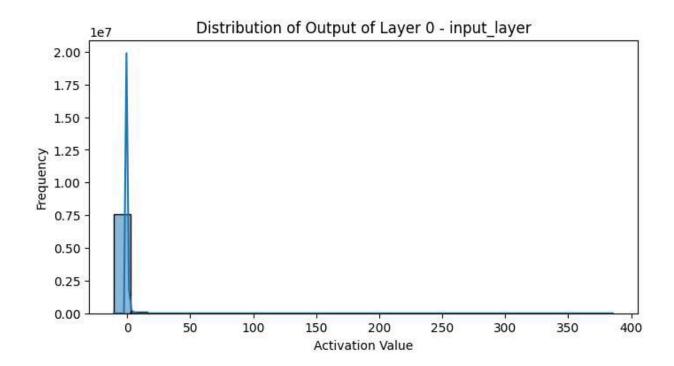
Training and Validation loss

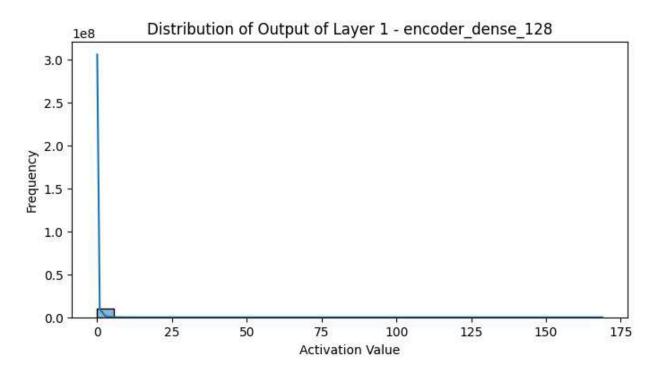


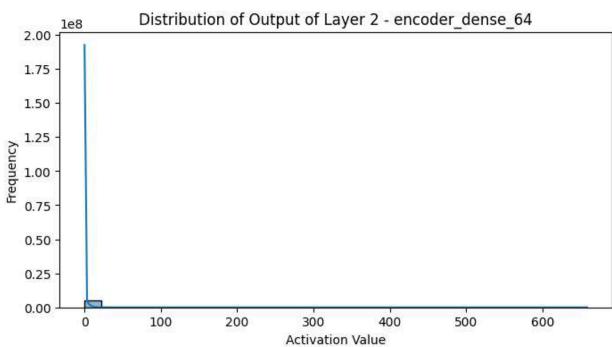
Weights of 15 extracted features for the first 100 samples

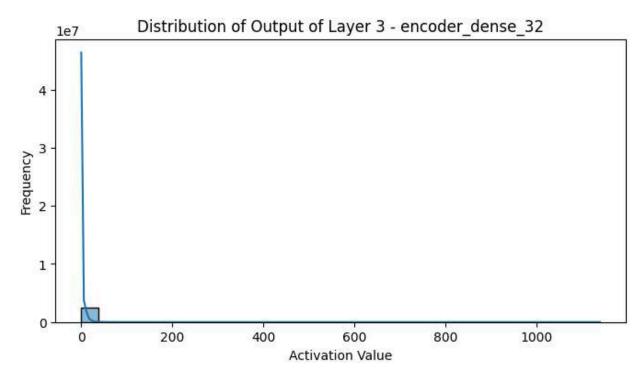


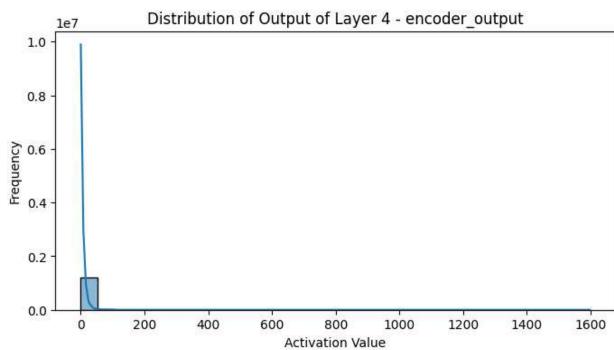
Layer Output Plots in autoencoder

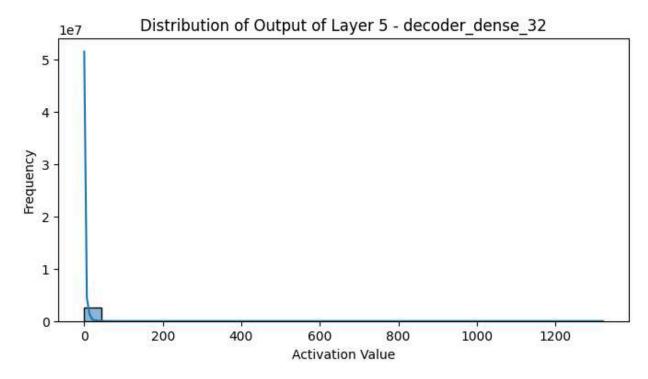


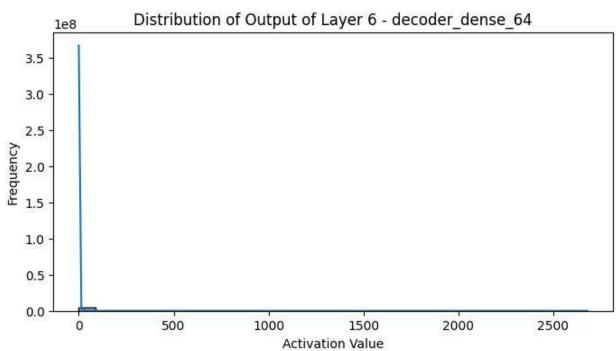


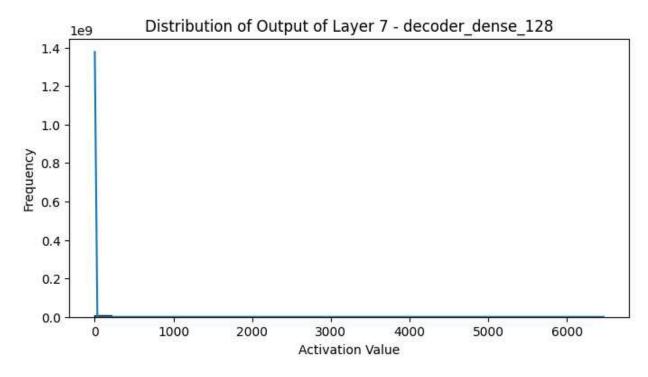


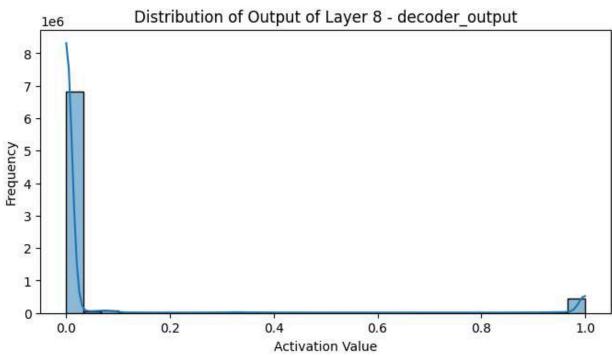












SVM Model trained on those 15 extracted features using 4 kernels

--- Training SVM with kernel: linear ---

Pickled SVM model saved as: svm_model_linear.pkl

Accuracy: 0.8674 Confusion Matrix: [[8893 1510] [1142 8455]]

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.89 | 0.85 | 0.87 | 10403 |
| 1 | 0.85 | 0.88 | 0.86 | 9597 |
| accuracy | | | 0.87 | 20000 |
| macro avg | 0.87 | 0.87 | 0.87 | 20000 |
| weighted avg | 0.87 | 0.87 | 0.87 | 20000 |

--- Training SVM with kernel: poly ---

Pickled SVM model saved as: svm_model_poly.pkl

Accuracy: 0.9056 Confusion Matrix: [[8847 1556] [332 9265]]

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.96 | 0.85 | 0.90 | 10403 |
| 1 | 0.86 | 0.97 | 0.91 | 9597 |
| accuracy | | | 0.91 | 20000 |
| macro avg | 0.91 | 0.91 | 0.91 | 20000 |
| weighted avg | 0.91 | 0.91 | 0.91 | 20000 |

--- Training SVM with kernel: rbf ---

Pickled SVM model saved as: svm_model_rbf.pkl

Accuracy: 0.9469 Confusion Matrix: [[9876 527] [535 9062]]

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.95 | 0.95 | 0.95 | 10403 |
| 1 | 0.95 | 0.94 | 0.94 | 9597 |
| accuracy | | | 0.95 | 20000 |
| macro avg | 0.95 | 0.95 | 0.95 | 20000 |
| weighted avg | 0.95 | 0.95 | 0.95 | 20000 |

--- Training SVM with kernel: sigmoid ---

Pickled SVM model saved as: svm_model_sigmoid.pkl

Accuracy: 0.7641 Confusion Matrix: [[7988 2415] [2303 7294]]

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 9 | 0.78 | 0.77 | 0.77 | 10403 |
| 1 | 0.75 | 0.76 | 0.76 | 9597 |
| accuracy | | | 0.76 | 20000 |
| macro avg | 0.76 | 0.76 | 0.76 | 20000 |
| weighted avg | 0.76 | 0.76 | 0.76 | 20000 |

PCA

```
PCA Component Loadings:
     full_url_length hostname_length ip_address_in_url dot_count \
            0.304053
                           0.279080
                                            0.000850 0.220172
PC2
            0.024396
                           -0.455301
                                            -0.002093 -0.188158
PC3
                           -0.128588
                                            0.002765 -0.287592
           0.014236
PC4
           0.190174
                           -0.043759
                                            -0.004768 -0.168799
          -8.823707
                          0.056629
                                            0.002499 -0.027297
           0.013734
                          -0.019609
                                            -0.000292 -0.001204
                                            -0.008269 0.022146
           0.071177
                          -0.122086
           0.048472
                          -0.070426
                                            -0.003945 -0.053668
PC8
            0.012567
                           0.107909
                                            -0.010204 0.044763
PC10
            0.000387
                           -0.061209
                                            -0.000717 -0.056522
     hyphen_count underscore_count slash_count question_mark_count \
         0.078555
                         0.060829
                                     0.095819
                                                         0.122430
PC2
         8.882895
                         0.029921
                                    0.117365
                                                        8.169966
                                    -0.044696
         0.008314
                         0.027744
                                                        -0.085202
        0.041807
                         0.091446
                                   -0.146808
                                                        -0.029979
        -0.002634
                         -0.005306
                                    -0.078632
                                                        -0.052837
PC5
       -0.002378
                         0.011353
                                    0.037627
                                                        0.005963
       -0.024517
                         0.003613
                                     8.310956
                                                        -0.005635
                                     0.022244
                                                        -0.003330
        0.037217
                         0.029973
PC9
        -0.100706
                        -0.051818
                                   -0.019788
                                                        -0.005489
        0.032595
                         0.022322
                                     0.182840
                                                        -0.201111
     equal_count at_count ... tld_length email_in_url \
        0.166285 0.018556 ... -0.020155
PC1
                                             0.010205
        0.209474 0.032092
                                0.005116
PC2
                                              0.020860
PC3
      -0.106784 -0.025646 ...
                               0.007342
                                             -0.011923
PC4
       0.068360 -0.030463
                                0.097085
                                             -0.030819
PC5
      -0.018276 0.040758
                               -0.009970
                                             0.002640
       -0.003917 0.033569
                                -8.028712
                                             0.006169
PC7
       -0.098423 -0.039209
                               0.065759
                                             -0.015373
       -0.065066 -0.077370
                               -0.119600
                                             -0.044746
       0.089932 0.040441
                                0.119989
                                             0.027208
PC18
       0.051801 -0.104956 ... -0.172308
                                             -0.102769
```

Feature Selection Methods

Mutual Information

| T 20 E W 1 | + |
|---------------------------|----------------------|
| Top 30 features by Mutual | |
| page_rank | 0.702414 0.701934 |
| domain_in_title | |
| google_index | 0.701579 |
| web_traffic | 0.701549 |
| mx_servers_count | 0.701454 |
| spf_record | 0.701269 |
| full_url_length | 0.199574 |
| average_word_length_url | 0.170532 |
| digit_ratio_full_url | 0.168474 |
| phish_hints | 0.159741 |
| word_count_url | 0.159658 |
| directory_length | 0.127919 |
| longest_word_url | 0.120555 |
| average_word_length_path | 0.116191 |
| number_of_subdomains | 0.114924 |
| parameters_length | 0.094753 |
| dot_count | 0.094540 |
| word_count_path | 0.092343 |
| hostname_length | 0.087688 |
| equal_count | 0.085184 |
| file_name_length | 0.084991 |
| longest_word_path | 0.080602 |
| number_of_parameters | 0.076075 |
| question_mark_count | 0.074773 |
| brand_in_path | 0.072622 |
| slash_count | 0.069799 |
| shortest_word_url | 0.067409 |
| ampersand_count | 0.066294 |
| www_occurrence | 0.065514 |
| digit_ratio_hostname | 0.060010 |
| dtype: float64 | |

```
--- Training SVM models using features from MutualInformation ---
Accuracy for MutualInformation with linear kernel: 99.3%
Accuracy for MutualInformation with poly kernel: 99.3%
Accuracy for MutualInformation with rbf kernel: 99.98%
Accuracy for MutualInformation with sigmoid kernel: 94.69%
```

Recursive Feature Elimination

```
Top features by RFE:

['dot_count', 'question_mark_count', 'equal_count', 'www_occurrence', 'com_occurrence', 'double_slash_occurrence', 'digit_ratio_full_url', 'digit_ratio_hostname', 'number _of_subdomains', 'prefix_suffix_hyphen', 'path_extension_check', 'char_repeat_path', 'shortest_word_path', 'longest_word_url', 'longest_word_path', 'average_word_length_ur l', 'average_word_length_path', 'phish_hints', 'brand_in_domain', 'brand_in_path', 'su spicious_tld', 'directory_length', 'tld_length', 'tld_present_in_parameters', 'mx_serv ers_count', 'spf_record', 'domain_in_title', 'web_traffic', 'google_index', 'page_ran k']
```

```
--- Training SVM models using features from RFE ---
Accuracy for RFE with linear kernel: 99.3%
Accuracy for RFE with poly kernel: 99.3%
Accuracy for RFE with rbf kernel: 99.98%
Accuracy for RFE with sigmoid kernel: 97.06%
```

ANOVA F Test

Top 30 features by ANOVA F-test: mx_servers_count 1.010210e+06 1.883286e+86 google_index spf_record 1.001989e+06 1.001906e+06 page_rank web_traffic 1.000525e+06 domain_in_title 9.964977e+05 phish_hints 2.512420e+04 full_url_length 1.950986e+04 1.838922e+84 word_count_url brand_in_path 1.406125e+84 question_mark_count 1.249869e+04 directory_length 1.215775e+04 shortest_word_url 1.162538e+84 longest_word_url 1.055542e+84 digit_ratio_full_url 1.005575e+04 equal_count 9.992471e+03 average_word_length_url 9 466164e+03 9.440946e+03 shortest_word_path longest_word_path 9.374884e+03 9.172405e+03 parameters_length slash_count 9.159346e+03 9.118613e+83 www_occurrence 8.965902e+03 word_count_path number_of_parameters 8.762073e+03 digit_ratio_hostname 7.455333e+03 dot_count 7.187682e+93 abnormal_subdomains 6.813939e+83 file_name_length 6.462871e+03 6.279237e+03 hyphen_count 6.068073e+03 tld_present_in_parameters

```
--- Training SVM models using features from ANOVAFtest ---
Accuracy for ANOVAFtest with linear kernel: 99.3%
Accuracy for ANOVAFtest with poly kernel: 99.3%
Accuracy for ANOVAFtest with rbf kernel: 99.98%
Accuracy for ANOVAFtest with sigmoid kernel: 98.48%
```

Extra Trees Classifier

```
Top 30 features by ExtraTreesClassifier importance:
google_index
                           0.223617
spf_record
                           0.173729
                           8.147566
page_rank
                           0.144260
domain_in_title
mx_servers_count
                           0.138384
web_traffic
                           0.104272
shortest_word_path
                           0.007581
brand_in_path
                           0.007471
www_occurrence
                           0.006159
tld_present_in_parameters 0.005530
digit_ratio_full_url
                           0.004560
phish_hints
                           0.004527
abnormal_subdomains
                           0.003573
shortest_word_url
                           0.003444
prefix_suffix_hyphen
                           0.002416
word_count_path
                           0.002393
parameters_length
                           0.002273
dot_count
                           0.002010
directory_length
                           0.001453
                           8.001429
digit_ratio_hostname
brand_in_domain
                           0.001411
full_url_length
                           8.001125
com_occurrence
                           0.001019
number_of_subdomains
                           0.000916
question_mark_count
                           0.000874
hostname_length
                           0.000863
number_of_parameters
                           0.000817
path_extension_check
                           0.000795
char_repeat_url
                           0.000759
slash_count
                           0.000702
dtype: float64
```

```
--- Training SVM models using features from ExtraTrees ---
Accuracy for ExtraTrees with linear kernel: 99.3%
Accuracy for ExtraTrees with poly kernel: 99.3%
Accuracy for ExtraTrees with rbf kernel: 99.98%
Accuracy for ExtraTrees with sigmoid kernel: 94.51%
```

Chi Square Test

```
Top 30 features by Chi-Square Test:
google_index
                           38975.140248
                           38957.575044
mx_servers_count
web_traffic
                            38893.887789
page_rank
                           38888.499609
spf_record
                            38882.572700
domain_in_title
                          38864.145400
                            9199.908797
brand_in_path
abnormal_subdomains
                           5739.655359
tld_present_in_parameters
                           5349.450499
                            4853.429174
phish_hints
prefix_suffix_hyphen
                           4264.781624
digit_ratio_full_url
                            2054.814452
brand_in_domain
                            1995.166222
digit_ratio_hostname
                            1660.688549
www_occurrence
                             1178.962170
equal_count
                             1101.426169
random_domain_indicator
                             1020.785938
path_extension_check
                             927.698783
number_of_parameters
                            914.760927
question_mark_count
                             778.622868
suspicious_tld
                             680.207655
ampersand_count
                             647.204417
                             571.520467
hyphen_count
parameters_length
                             570.311470
http_occurrence
                             493.320645
full_url_length
hostname_length
                            448.826311
                             420 407214
word_count_url
shortest_word_url
                             393.252918
slash_count
                             316.363297
dtype: float64
```

```
--- Training SVM models using features from ChiSquare ---
Accuracy for ChiSquare with linear kernel: 99.3%
Accuracy for ChiSquare with poly kernel: 99.3%
Accuracy for ChiSquare with rbf kernel: 99.3%
Accuracy for ChiSquare with sigmoid kernel: 99.82%
```

IKS - Vedic Maths

Vedic Multiplication (Urdhva-Tiryagbhyam)

Urdhva - Tiryagbhyam

Case 1: Multiplication of two digit numbe

Ex: Multiply 14 by 12 i.e. 14 X 12

1 6 8

Ans: 168

1.
$$4 \times 2 = 8$$

2.
$$(1x2) + (4x1)$$

2 + 4 = 6

$$3.1 \times 1 = 1$$

Example: Multiplying 23 × 45

Step 1 - Write the Numbers as Digits:

 $23 \rightarrow$ digits: 2 and 3 $45 \rightarrow$ digits: 4 and 5

Step 2 – Multiply the Right-most Digits:

Multiply 3 (from 23) by 5 (from 45): 3×5=15

Write down 5 and carry over 1.

Step 3 – Cross-Multiply and Add:

Multiply cross-wise:

$$(2 \times 5) + (3 \times 4) = 10 + 12 = 22$$

Add the carried over 1:

$$22 + 1 = 23$$

Write down the unit digit 3 and carry over 2.

Step 4 – Multiply the Left-most Digits:

Multiply 2 (from 23) by 4 (from 45):

2×4=8

Add the carry 2: 8+2=10

Write down 10 (which gives the remaining digits).

Step 5 – Combine the Results:

The digits (from left to right) become 10, 3, 5

When you combine them (taking care of any place-value adjustments), the final product is 1035.

Matrix Dot Product Using Vedic Multiplication

result[i,j]= \sum vedic_multiply(A[i,k],B[k,j])