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
```

Drop highly correlated Columns with corr > 0.9

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

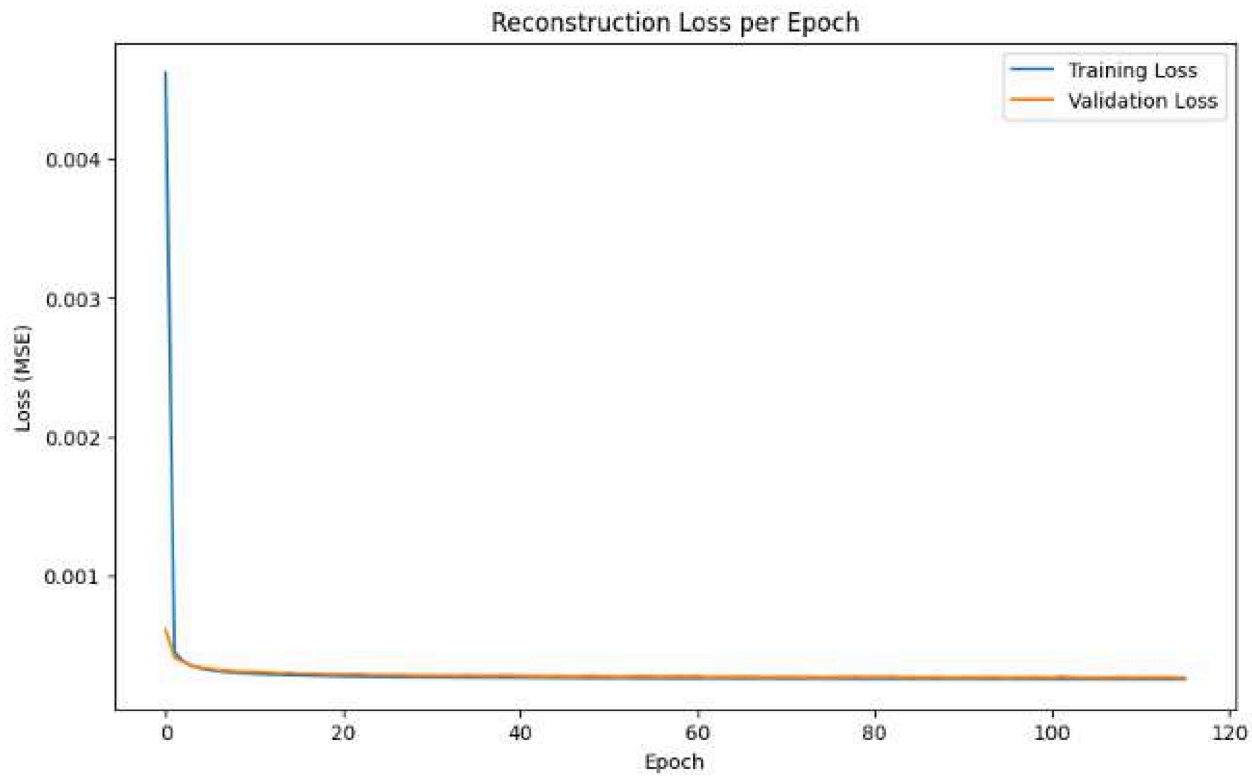
```
Model: "model_28"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_15 (InputLayer)       [(None, 97)]              0

 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.

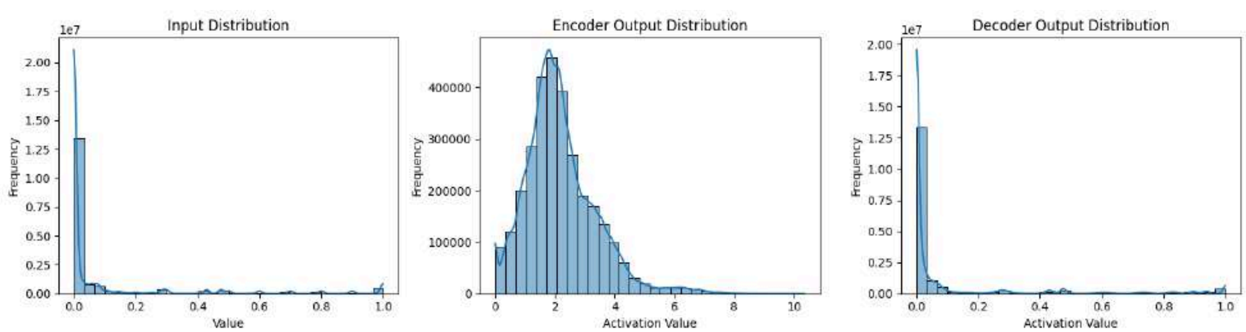
Trained till 128 epochs with validation loss of 0.0002

```
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

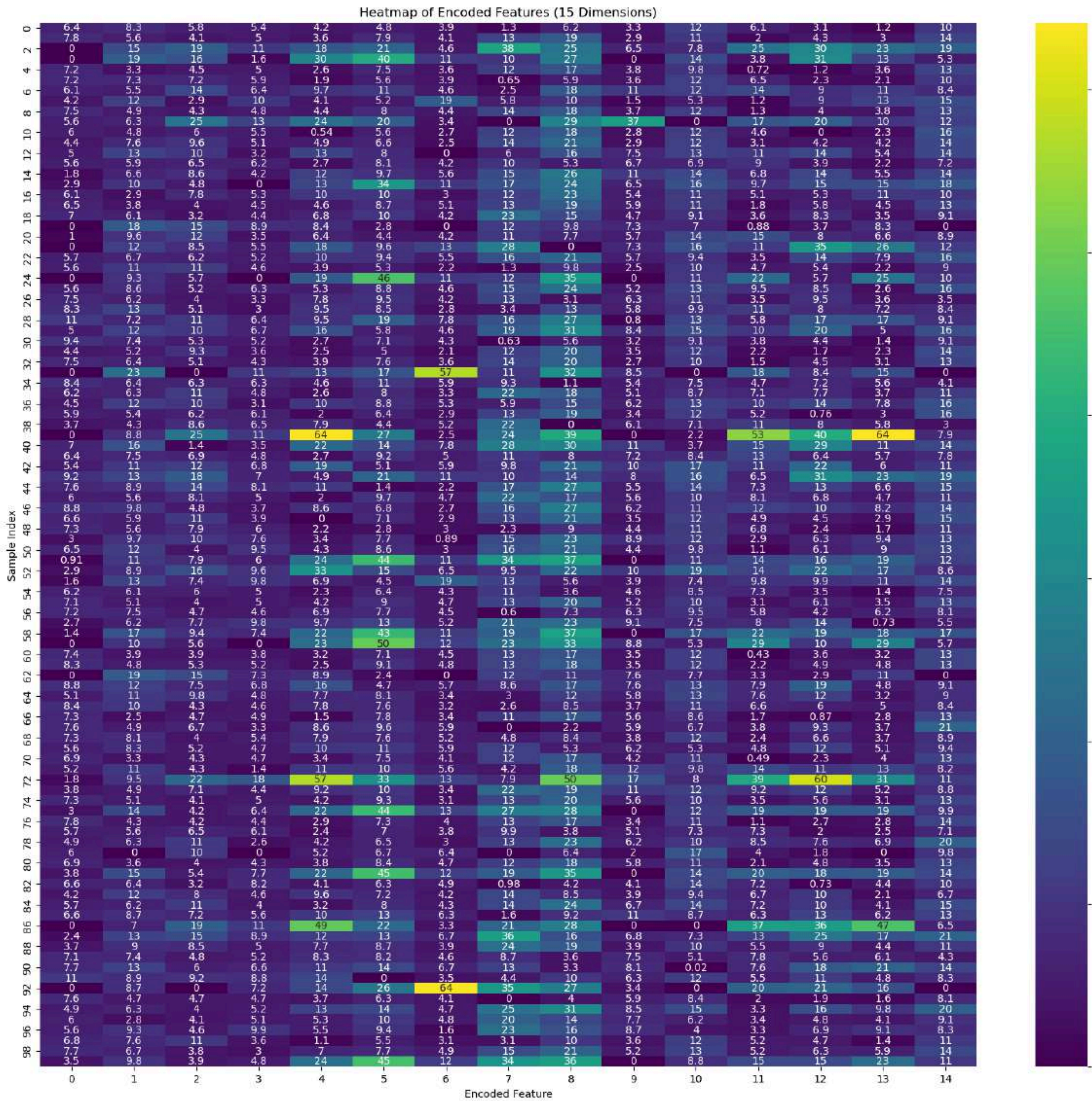


Plots of input data to each layer



Weights of Extracted Features for the first 100 samples

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



Heatmap of Encoded Features (15 Dimensions)
Sample Index
Encoded Feature

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

           0       0.93      0.95      0.94     10403
           1       0.94      0.93      0.93      9597

    accuracy                           0.94     20000
   macro avg       0.94      0.94      0.94     20000
weighted avg       0.94      0.94      0.94     20000
```

```
--- Training SVM with kernel: poly ---
Accuracy: 0.963
Confusion Matrix:
 [[10041   362]
 [  378  9219]]
Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.97      0.96     10403
           1       0.96      0.96      0.96      9597

    accuracy                           0.96     20000
   macro avg       0.96      0.96      0.96     20000
weighted avg       0.96      0.96      0.96     20000
```

```
--- Training SVM with kernel: rbf ---
Accuracy: 0.97225
Confusion Matrix:
 [[10199   204]
 [  351  9246]]
Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.98      0.97     10403
           1       0.98      0.96      0.97      9597

    accuracy                           0.97     20000
   macro avg       0.97      0.97      0.97     20000
weighted avg       0.97      0.97      0.97     20000
```

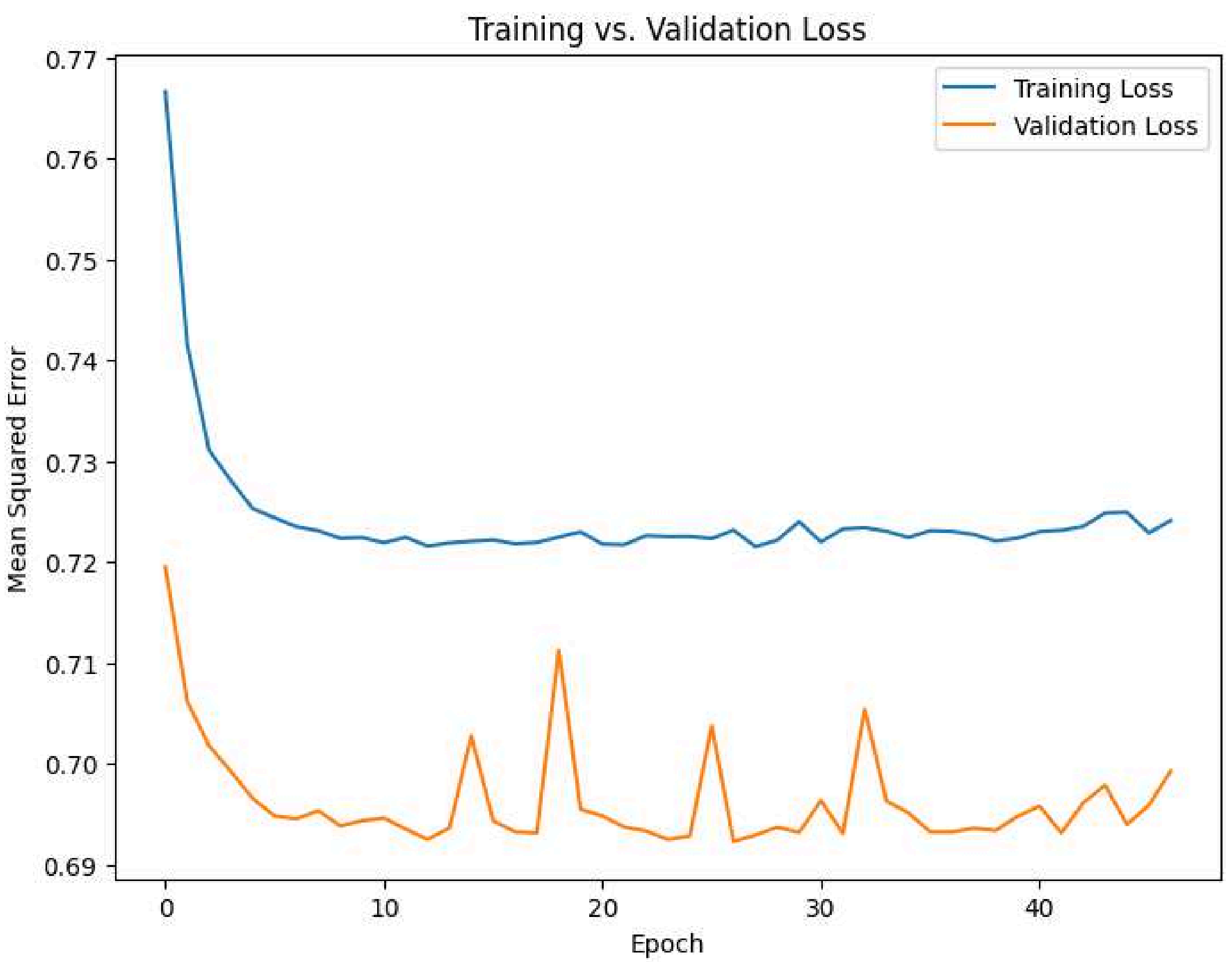
```
--- Training SVM with kernel: sigmoid ---
Accuracy: 0.3808
Confusion Matrix:
 [[4199 6204]
 [6180 3417]]
Classification Report:
              precision    recall  f1-score   support

           0       0.40      0.40      0.40     10403
           1       0.36      0.36      0.36      9597

    accuracy                           0.38     20000
   macro avg       0.38      0.38      0.38     20000
weighted avg       0.38      0.38      0.38     20000
```

Complex Autoencoder architecture

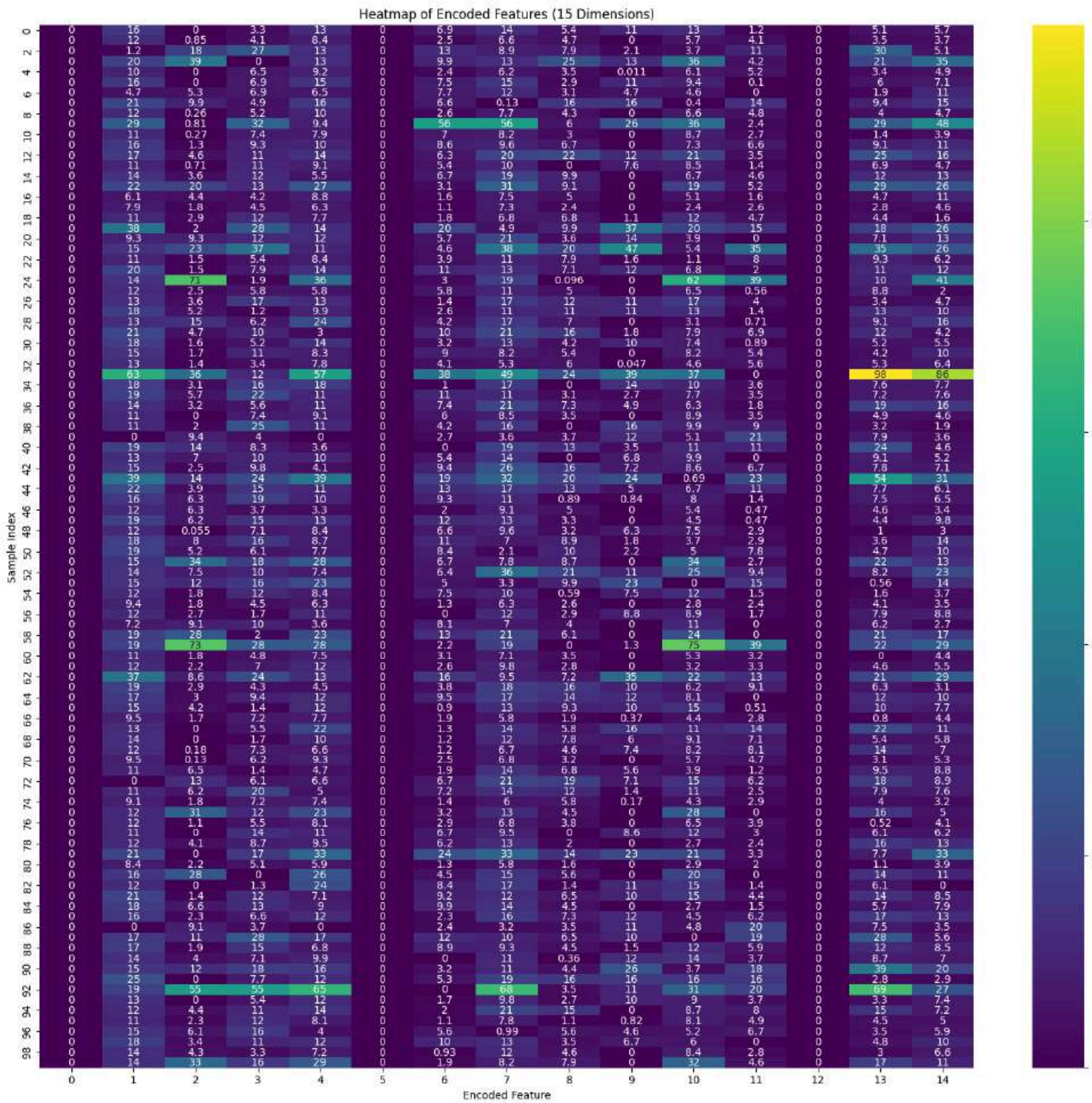
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 97)	0
encoder_dense_128 (Dense)	(None, 128)	12,544
encoder_dense_64 (Dense)	(None, 64)	8,256
encoder_dense_32 (Dense)	(None, 32)	2,080
encoder_output (Dense)	(None, 15)	495
decoder_dense_32 (Dense)	(None, 32)	512
decoder_dense_64 (Dense)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

```
3995/4000 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7570Epoch 046 - loss: 0.7229 - val_lo
ss: 0.6959
4000/4000 ━━━━━━━━━━━━━━━━━━━━ 9s 2ms/step - loss: 0.7570 - val_loss: 0.6959
Epoch 47/500
3985/4000 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7523Epoch 047 - loss: 0.7241 - val_lo
ss: 0.6993
4000/4000 ━━━━━━━━━━━━━━━━━━━━ 9s 2ms/step - loss: 0.7522 - val_loss: 0.6993
```

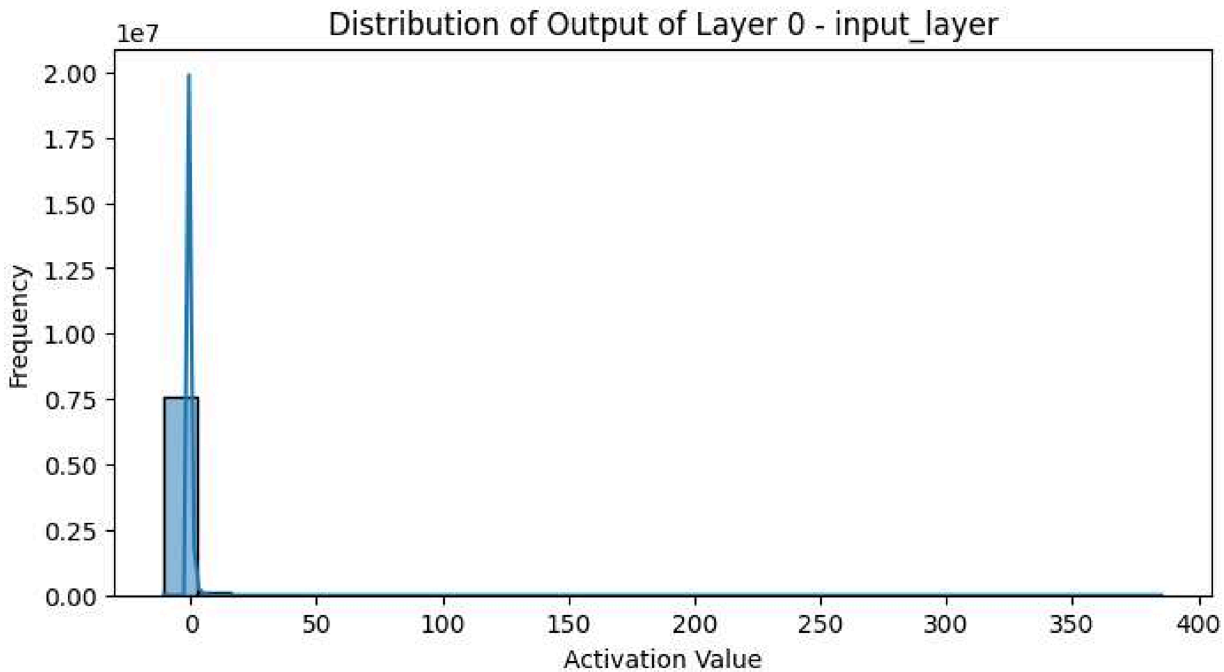
Training and Validation loss

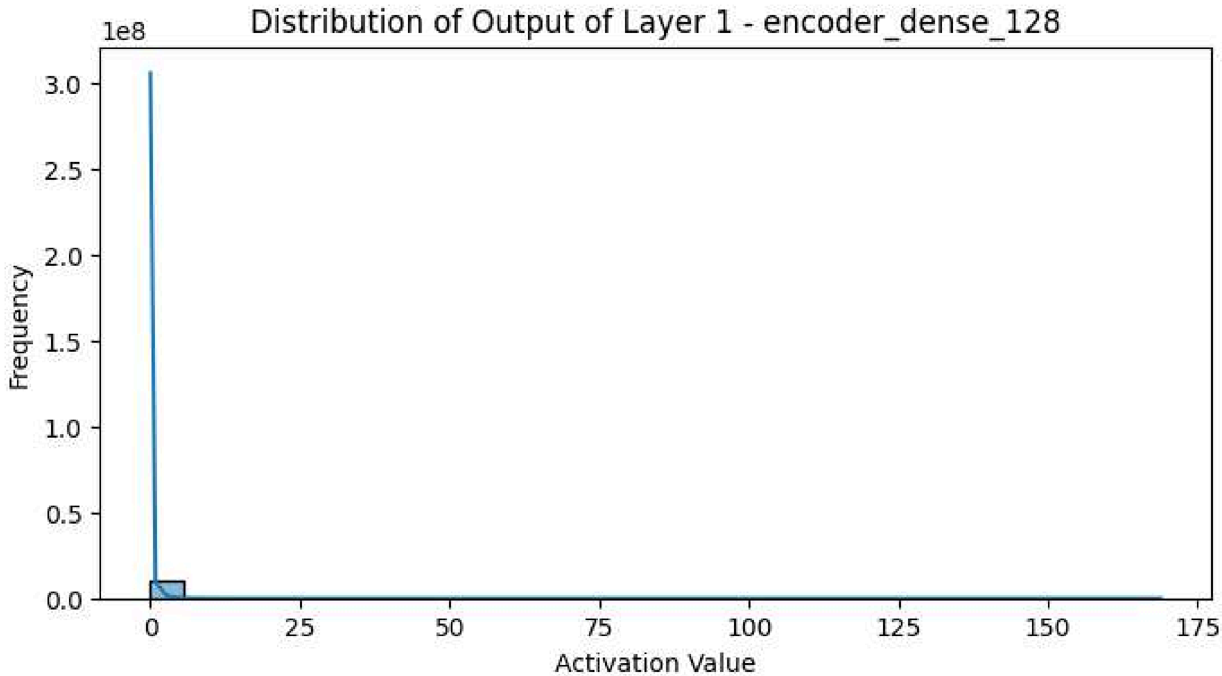


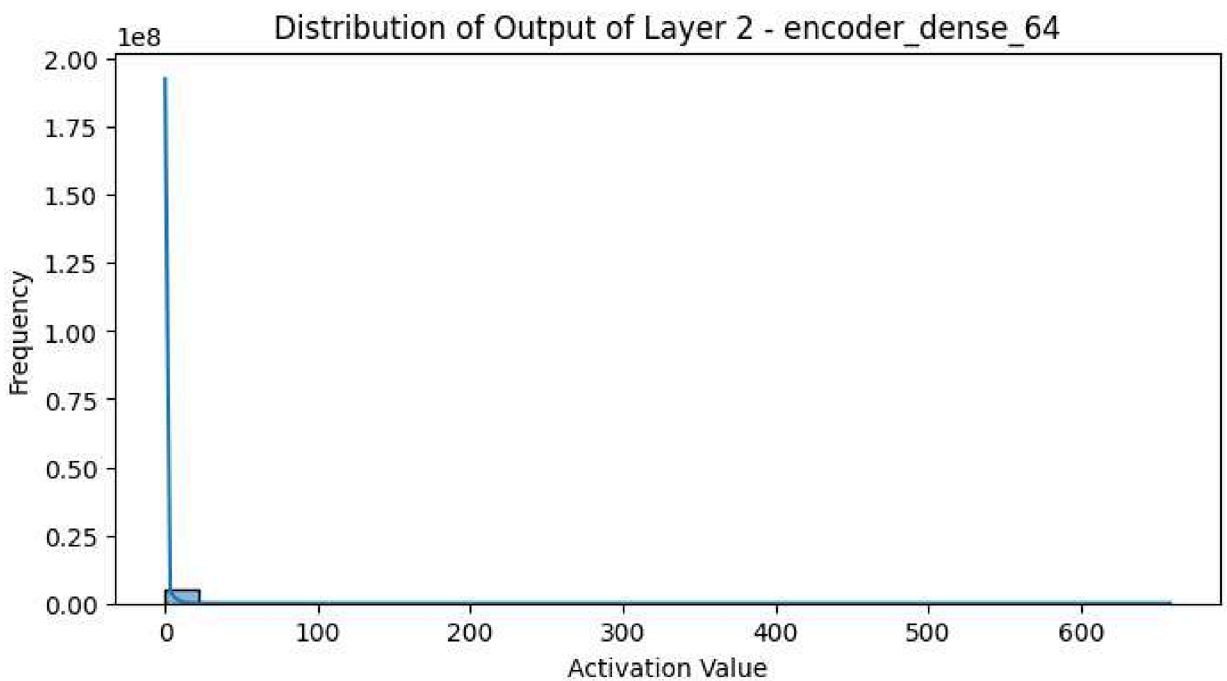
Weights of 15 extracted features for the first 100 samples

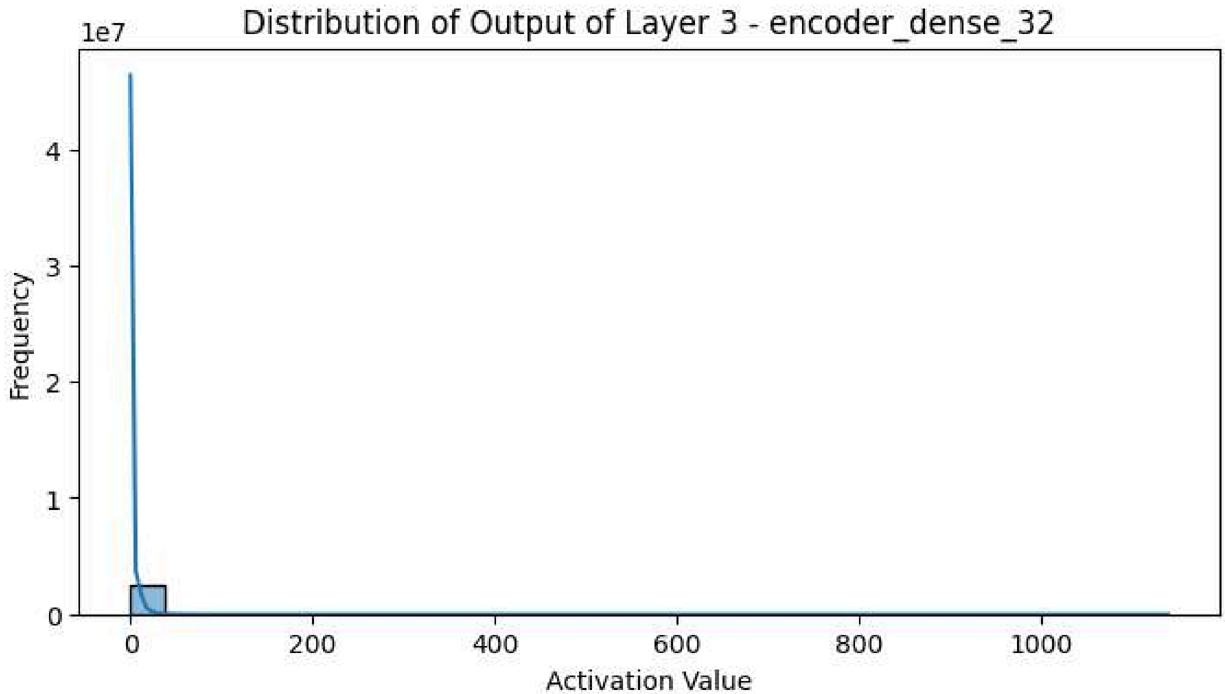


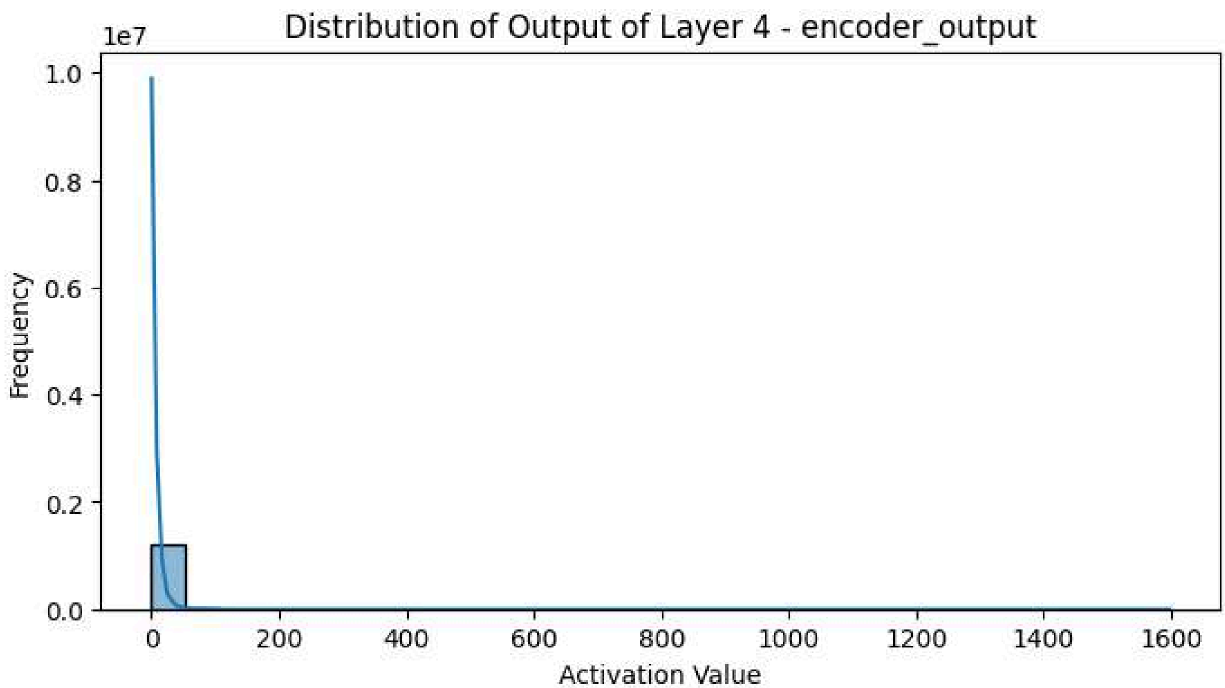
Layer Output Plots in autoencoder

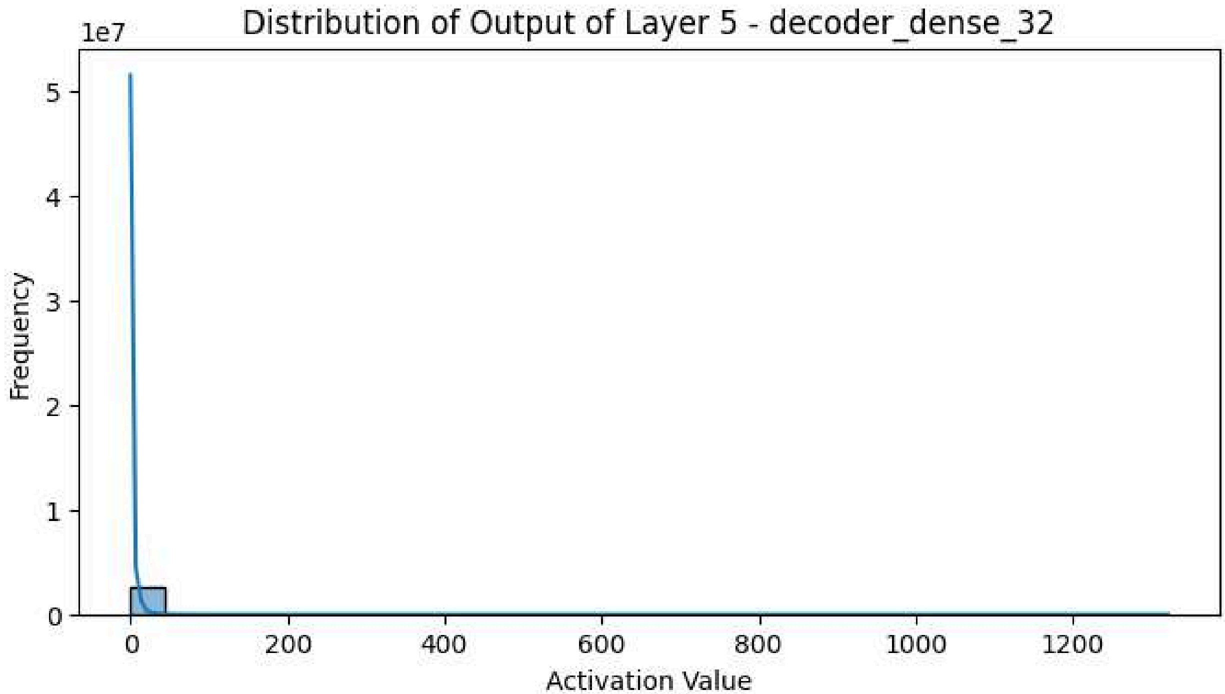



Distribution of Output of Layer 1 - encoder_dense_128
1e8
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency
0
25
50
75
100
125
150
175
Activation Value

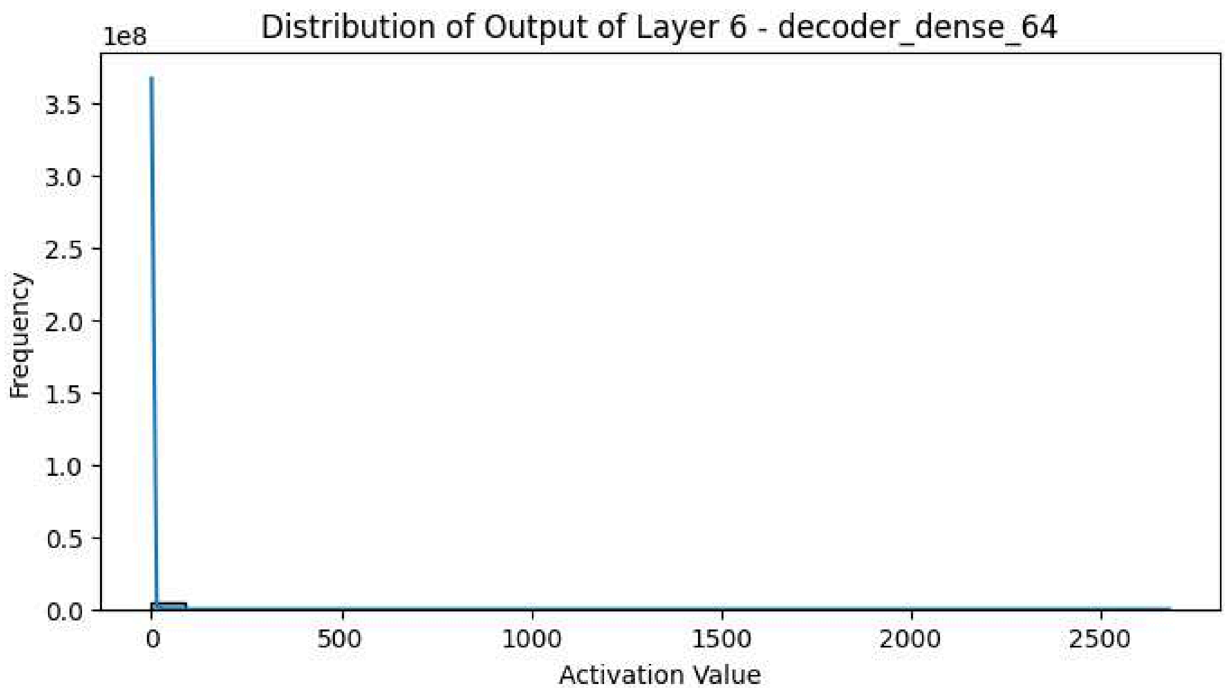

Distribution of Output of Layer 2 - encoder_dense_64
1e8
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Frequency
0
100
200
300
400
500
600
Activation Value


Distribution of Output of Layer 3 - encoder_dense_32
1e7
4
3
2
1
0
Frequency
0
200
400
600
800
1000
Activation Value

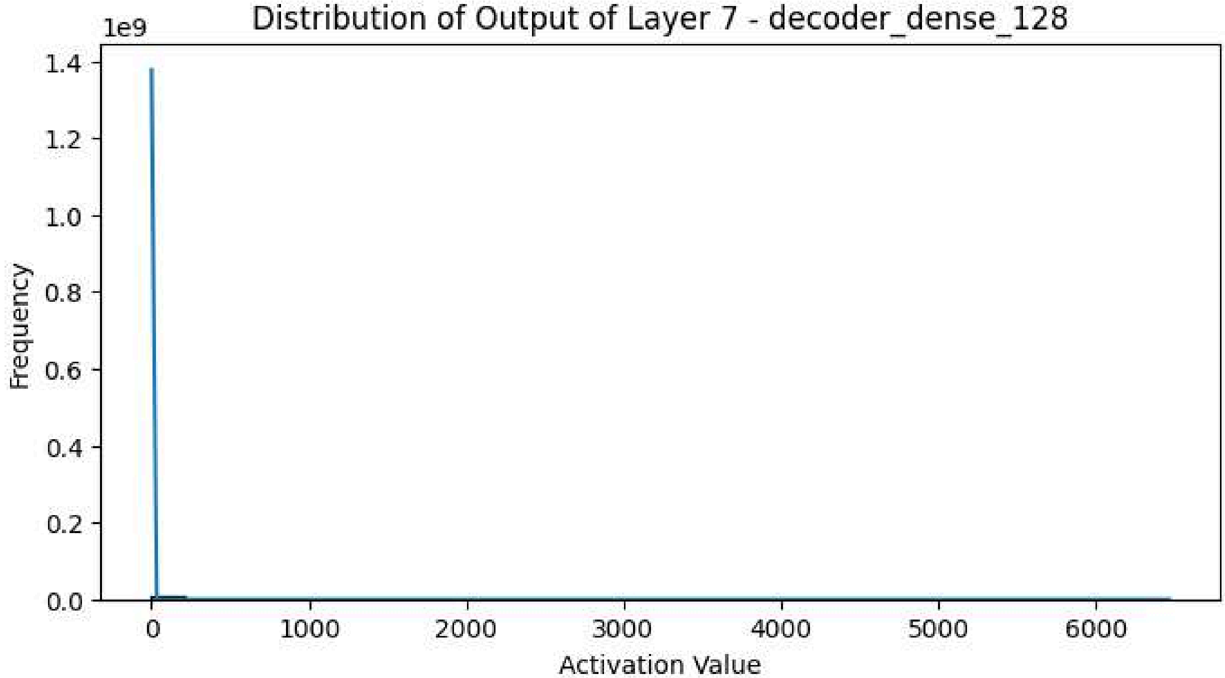

Distribution of Output of Layer 4 - encoder_output
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
0
200
400
600
800
1000
1200
1400
1600
Activation Value

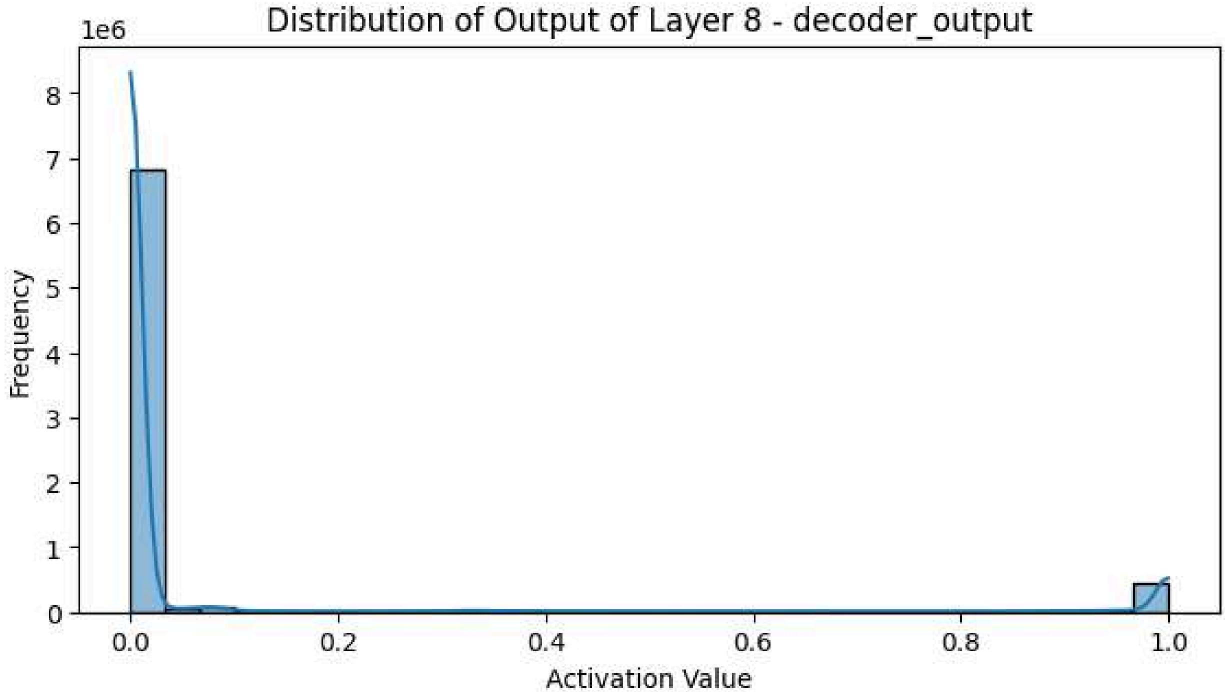


Distribution of Output of Layer 5 - decoder_dense_32
1e7
Frequency
Activation Value
0
1
2
3
4
5
0
200
400
600
800
1000
1200



Distribution of Output of Layer 6 - decoder_dense_64
1e8
Frequency
Activation Value
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
500
1000
1500
2000
2500





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

           0       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  \
PC1          0.304053         0.279080           0.000850   0.220172
PC2          0.024396        -0.455301          -0.002093  -0.188158
PC3          0.014236        -0.128588           0.002765  -0.287592
PC4          0.190174        -0.043759          -0.004768  -0.168799
PC5         -0.023707         0.056629           0.002499  -0.027297
PC6          0.013734        -0.019609          -0.000292  -0.001204
PC7          0.071177        -0.122086          -0.008269   0.022146
PC8          0.048472        -0.070426          -0.003945  -0.053668
PC9          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  \
PC1       0.078555          0.060829     0.095819             0.122430
PC2       0.002095          0.029921     0.117365             0.169966
PC3       0.008314          0.027744    -0.044696            -0.085202
PC4       0.041807          0.091446    -0.146808            -0.029979
PC5      -0.002634         -0.005306    -0.078632            -0.052837
PC6      -0.002378          0.011353     0.037627             0.005963
PC7      -0.024517          0.003613     0.310956            -0.005635
PC8       0.037217          0.029973     0.022244            -0.003330
PC9      -0.100706         -0.051818    -0.019788            -0.005489
PC10      0.032595          0.022322     0.182840            -0.201111

      equal_count  at_count  ...  tld_length  email_in_url  \
PC1      0.166285  0.018556  ...   -0.020155      0.010205
PC2      0.209474  0.032092  ...    0.005116      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
PC6     -0.003917  0.033569  ...   -0.028712      0.006169
PC7     -0.098423 -0.039209  ...    0.065759     -0.015373
PC8     -0.065066 -0.077370  ...   -0.119600     -0.044746
PC9      0.089932  0.040441  ...    0.119989      0.027208
PC10     0.051801 -0.104956  ...   -0.172308     -0.102769

      tld_present_in_parameters  number_of_parameters  mx_servers_count  \
```

Feature Selection Methods

Mutual Information

```
Top 30 features by Mutual Information:
page_rank                    0.702414
domain_in_title              0.701934
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             0.084001
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_occurrenc
e', '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
google_index                 1.003206e+06
spf_record                   1.001989e+06
page_rank                    1.001906e+06
web_traffic                  1.000525e+06
domain_in_title              9.964977e+05
phish_hints                  2.512420e+04
full_url_length              1.950986e+04
word_count_url               1.838922e+04
brand_in_path                1.406125e+04
question_mark_count          1.249869e+04
directory_length             1.215775e+04
shortest_word_url            1.162530e+04
longest_word_url             1.055542e+04
digit_ratio_full_url         1.005575e+04
equal_count                  9.992471e+03
average_word_length_url      9.466164e+03
shortest_word_path           9.440946e+03
longest_word_path            9.374884e+03
parameters_length            9.172405e+03
slash_count                  9.159346e+03
www_occurrence               9.118613e+03
word_count_path              8.965902e+03
number_of_parameters         8.762073e+03
digit_ratio_hostname         7.455333e+03
dot_count                    7.187682e+03
abnormal_subdomains          6.813939e+03
file_name_length             6.462871e+03
hyphen_count                 6.279237e+03
tld_present_in_parameters    6.068073e+03
dtype: float64
```

```
--- 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
page_rank                     0.147566
domain_in_title               0.144260
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
digit_ratio_hostname          0.001429
brand_in_domain               0.001411
full_url_length               0.001125
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%
```

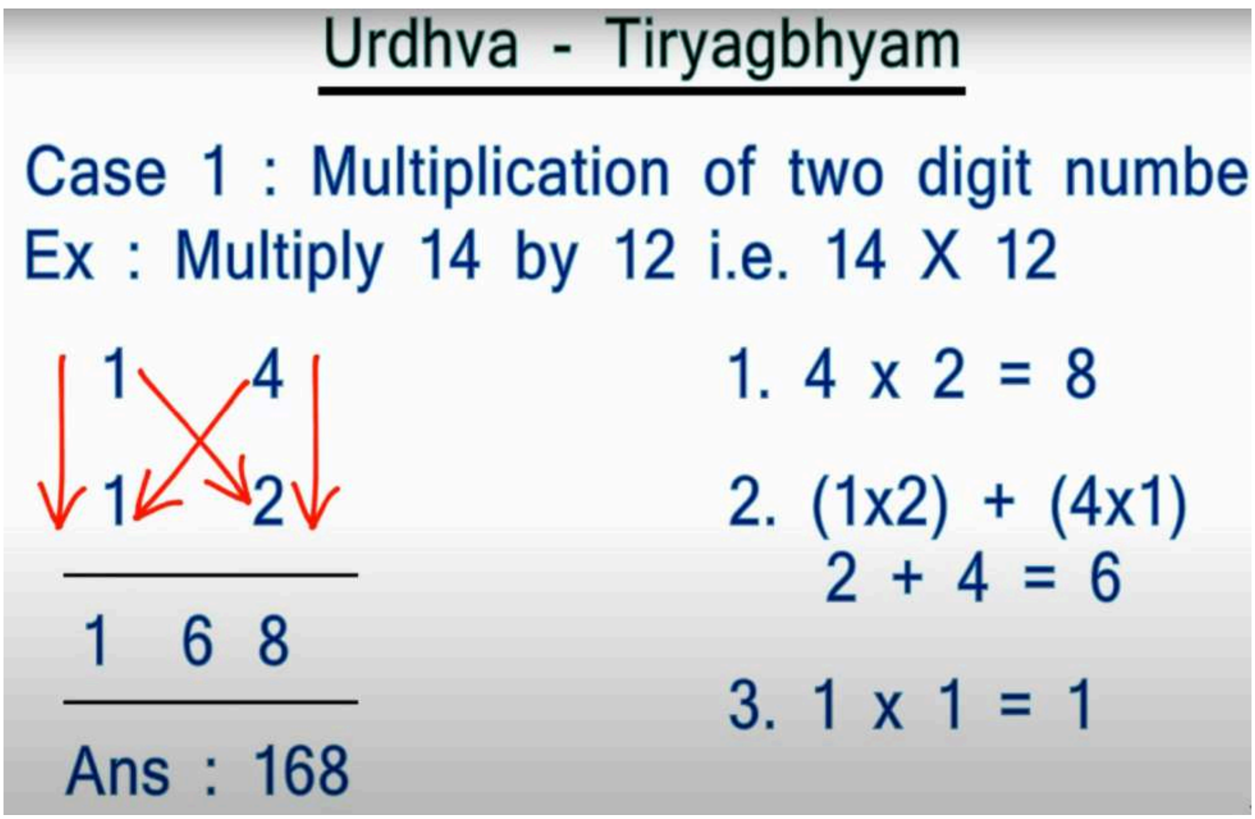
Chi Square Test

```
Top 30 features by Chi-Square Test:
google_index                  38975.140248
mx_servers_count              38957.575044
web_traffic                   38893.887709
page_rank                     38888.499609
spf_record                    38882.572700
domain_in_title               38864.145400
brand_in_path                  9199.908797
abnormal_subdomains            5739.655359
tld_present_in_parameters      5349.450499
phish_hints                    4853.429174
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        1020.785930
path_extension_check            927.698783
number_of_parameters            914.760927
question_mark_count             778.622868
suspicious_tld                  680.207655
ampersand_count                 647.204417
hyphen_count                    571.520467
parameters_length               570.311470
http_occurrence                 493.320645
full_url_length                 469.115935
hostname_length                 448.826311
word_count_url                  420.407214
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)



Example: Multiplying 23 × 45

Step 1 – Write the Numbers as Digits:

23 → digits: 2 and 3
45 → digits: 4 and 5

Step 2 – Multiply the Right-most Digits:

Multiply 3 (from 23) by 5 (from 45): $3 \times 5 = 15$

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):

$2 \times 4 = 8$

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

$$result[i,j] = \sum vedic\_multiply(A[i,k], B[k,j])$$