Host based intrusion detection systems

Isha Mangurkar
Tanmai Gajula
A Ramya Keerthana

Types of intrusion detection systems

1. Anomaly detection : trained with normal packets and abnormal packets. System predicts which class the next packet will be placed in based on a definition of normal class.

Two ways of doing this is using SVM and Neural Networks.

2. Misuse detection : Learn signature attacks and train on every possible module. Categories of intrusion types imply multi-class SVM or NN use.

Need for preprocessing

1. Large quantity of data points are present in the dataset.
2. Need to reduce the dataset to a set of usable data points.
3. Can be done by reducing the number of features used for classification to only the important ones.
4. This process requires pre-processing of data.
5. From 41 features, using fast feature reduction, we need to derive a set of most important features for classification.
6. Advantage of this method : speed.

Fast Feature Reduction

1. In this method, there are K=5 classes.
2. For each class, we add all data points which are in the same class, feature by feature.
3. We calculate the scattering of points along each feature.
4. For high turbulence in feature d among all classes, we can say that that feature has high power in the classification.
5. Sorting the matrix in ascending order, we choose t features for classification.

SVM Intrusion Detection System

- The data is first partitioned into two classes: normal and attack, where attack represents a collection of 22 different attacks belonging to the four classes DOS, R2L, U2R, Probing.
- **Training** : The SVMs are trained with normal and intrusive data.
    - Each point is located in the n-dimensional space, with each dimension corresponding to a feature of the data point.
    - RBF(Radial Bias Function) Kernel function is used.
    - There are four experiments performed: The training sets consists of 7000 points and 10000 points for the entire feature set of 41 features as well as reduced feature set of 13 features.
- **Testing** : SVMs are used to differentiate intrusions and normal activities. In the four experiments, the testing set consists of 7000 points and 40000 points for the two cases of entire and reduces feature sets.

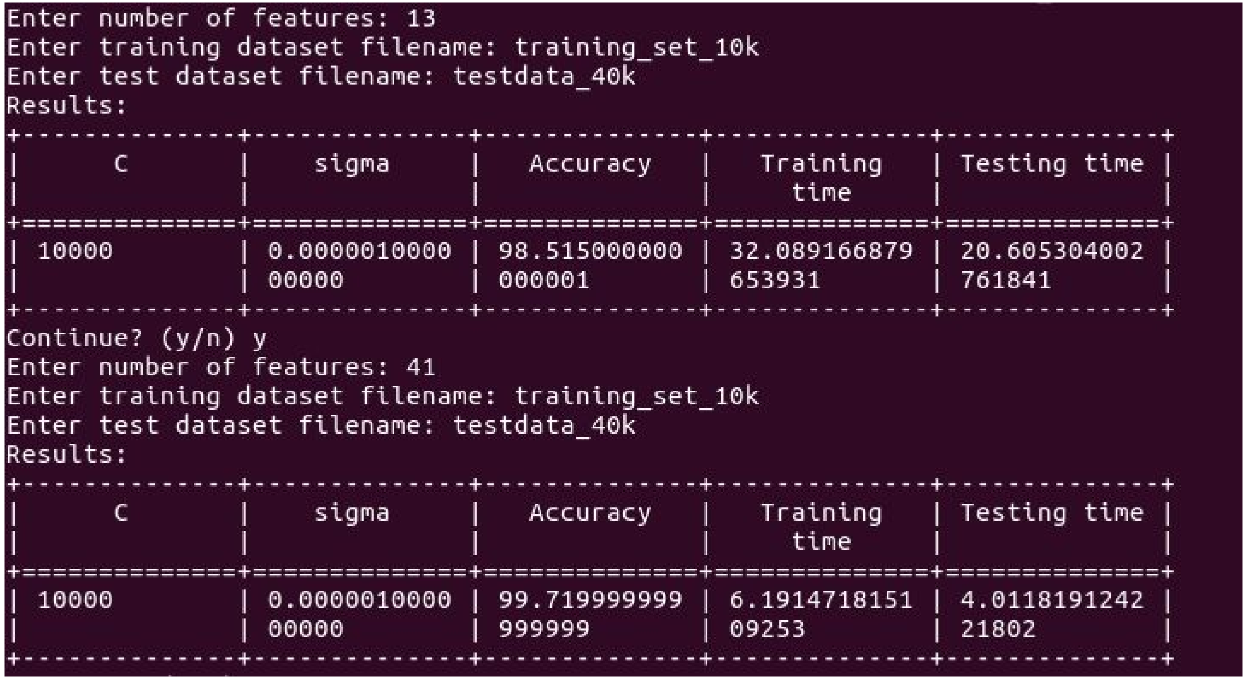
Results with SVM

Experiment 1:

Training set: 10000 points
Testing set: 40000 points
Features: 13

Experiment 2:

Training set: 10000 points
Testing set: 40000 points
Features: 41



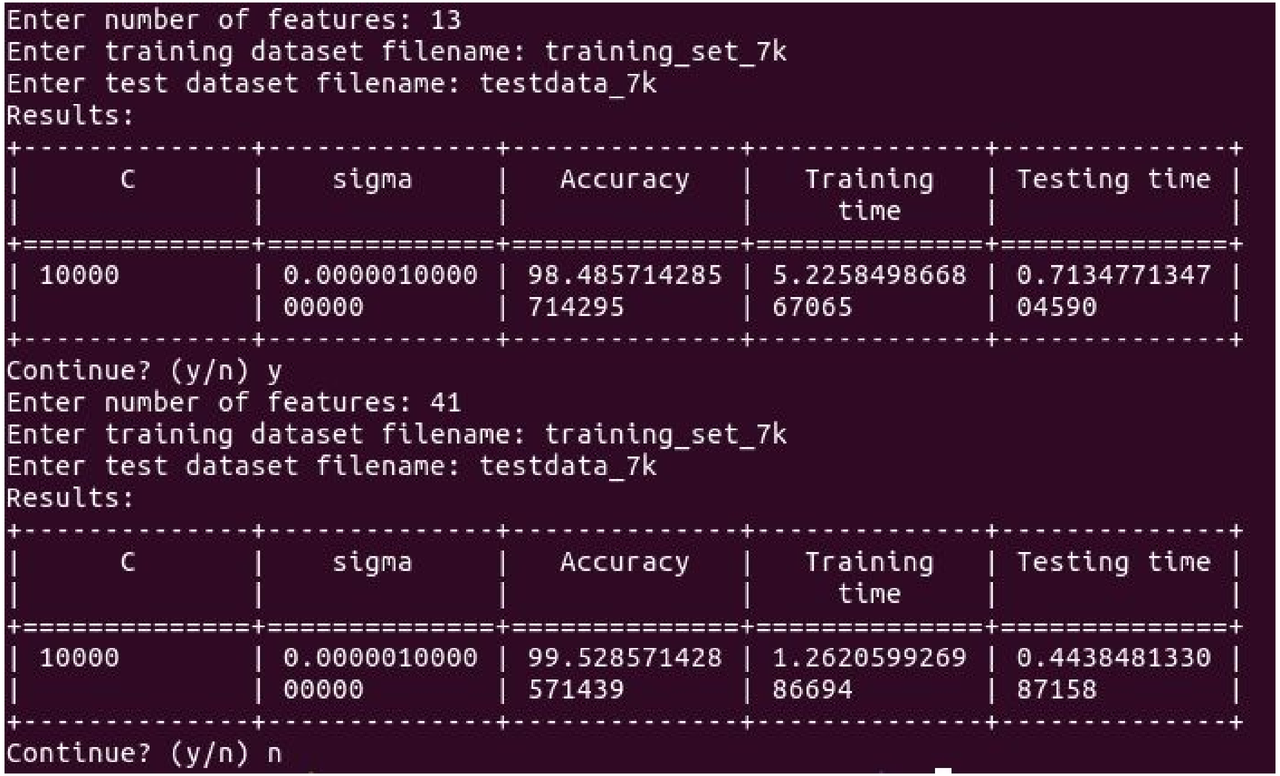
```
Enter number of features: 13
Enter training dataset filename: training_set_10k
Enter test dataset filename: testdata_40k
Results:
+-------------+-------------+-------------+-------------+-------------+
|      C      |    sigma    |  Accuracy   |  Training   | Testing time |
|             |             |             |    time     |             |
+=============+=============+=============+=============+=============+
| 10000       | 0.0000010000 | 98.515000000 | 32.089166879 | 20.605304002 |
|             | 00000       | 000001      | 653931      | 761841      |
+-------------+-------------+-------------+-------------+-------------+
Continue? (y/n) y
Enter number of features: 41
Enter training dataset filename: training_set_10k
Enter test dataset filename: testdata_40k
Results:
+-------------+-------------+-------------+-------------+-------------+
|      C      |    sigma    |  Accuracy   |  Training   | Testing time |
|             |             |             |    time     |             |
+=============+=============+=============+=============+=============+
| 10000       | 0.0000010000 | 99.719999999 | 6.1914718151 | 4.0118191242 |
|             | 00000       | 999999      | 09253       | 21802       |
+-------------+-------------+-------------+-------------+-------------+
```

Experiment 3:

Training set: 7000 points
Test set: 7000 points
Features:13

Experiment 4:

Training set: 7000 points
Test set: 7000 points
Features: 41



```
Enter number of features: 13
Enter training dataset filename: training_set_7k
Enter test dataset filename: testdata_7k
Results:
+-------------+-------------+-------------+-------------+-------------+
|      C      |    sigma    |  Accuracy   |  Training   | Testing time |
|             |             |             |    time     |             |
+=============+=============+=============+=============+=============+
| 10000       | 0.0000010000 | 98.485714285 | 5.2258498668 | 0.7134771347 |
|             | 00000       | 714295      | 67065       | 04590       |
+-------------+-------------+-------------+-------------+-------------+
Continue? (y/n) y
Enter number of features: 41
Enter training dataset filename: training_set_7k
Enter test dataset filename: testdata_7k
Results:
+-------------+-------------+-------------+-------------+-------------+
|      C      |    sigma    |  Accuracy   |  Training   | Testing time |
|             |             |             |    time     |             |
+=============+=============+=============+=============+=============+
| 10000       | 0.0000010000 | 99.528571428 | 1.2620599269 | 0.4438481330 |
|             | 00000       | 571439      | 86694       | 87158       |
+-------------+-------------+-------------+-------------+-------------+
Continue? (y/n) n
```

Neural Network Intrusion Detection System

- **Training**:
- neural network is trained on different types of attacks and normal data. The input has 41 features and the output assumes one of two values: intrusion (22 different attack types), and normal data.
- The training of the neural networks was conducted using feed forward back propagation algorithm.
- Network A: 3-layer 41-50-40- 1 ; B: 3-layer, 13-50-40- 1; C: 3-layer 41-40-40-5; D:3-layer 13-40-40-5
- The purpose of having multiple networks is to find a suitable architecture that can detect at a faster speed with low error rate, minimizing false positives and false negatives.

Testing:

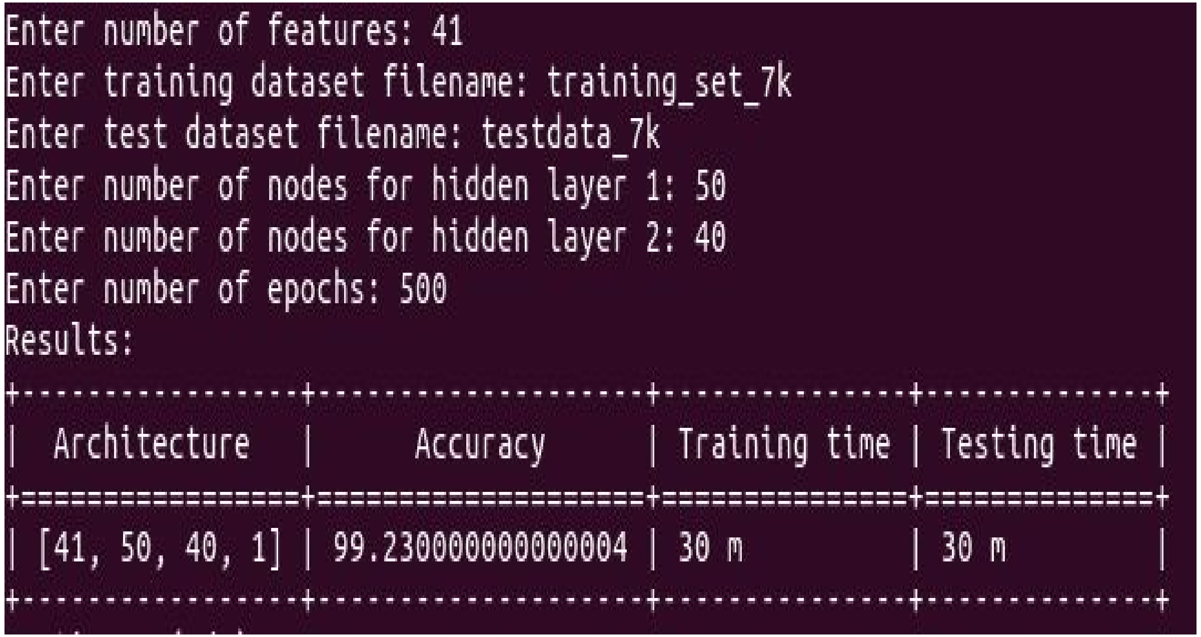
- The testing again consisting of 7000 data points with 41 features and 13 features.
- We have different feed-forward, multi-layer neural network architectures.

Experiment 1:(2 class)

Training set: 7000 points

Testing set: 7000 points

Features: 41



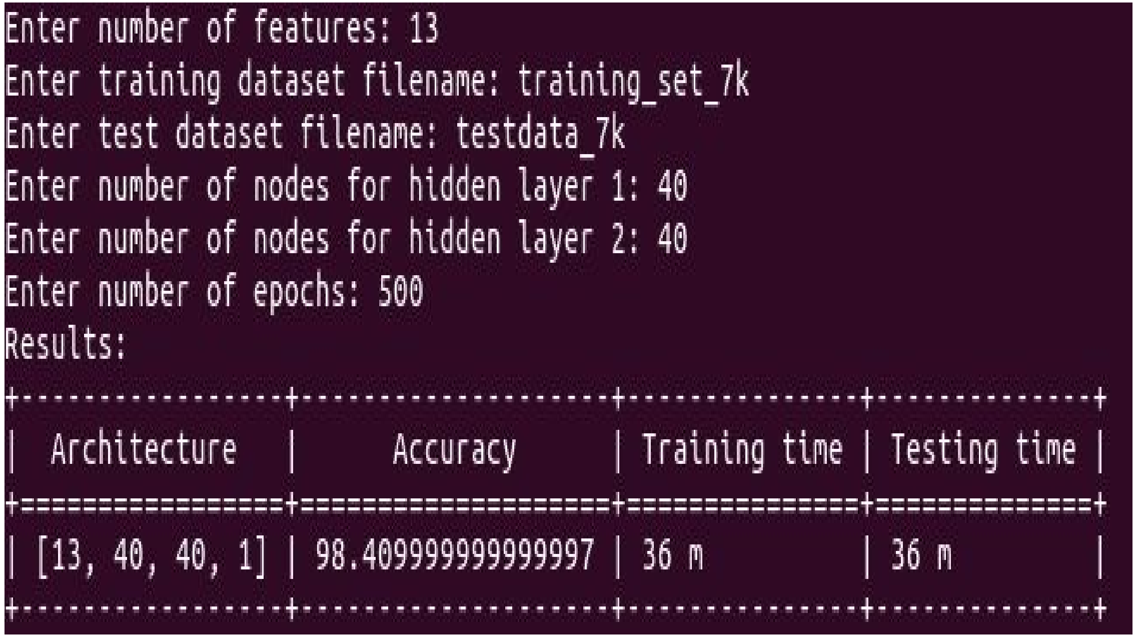
```
Enter number of features: 41
Enter training dataset filename: training_set_7k
Enter test dataset filename: testdata_7k
Enter number of nodes for hidden layer 1: 50
Enter number of nodes for hidden layer 2: 40
Enter number of epochs: 500
Results:
+-----------------+--------------------+--------------+--------------+
|  Architecture   |      Accuracy      | Training time | Testing time |
+=================+====================+==============+==============+
| [41, 50, 40, 1] | 99.230000000000004 | 30 m         | 30 m         |
+-----------------+--------------------+--------------+--------------+
```

Experiment 2(2 class)

Training set: 7000 points

Testing set: 7000 points

Features: 13



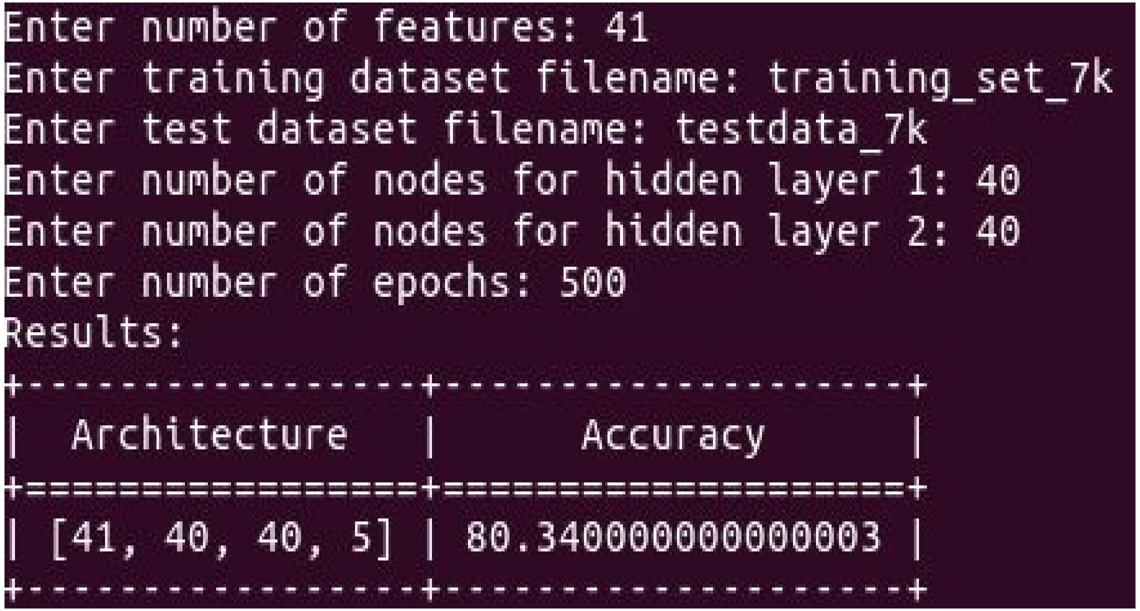
```
Enter number of features: 13
Enter training dataset filename: training_set_7k
Enter test dataset filename: testdata_7k
Enter number of nodes for hidden layer 1: 40
Enter number of nodes for hidden layer 2: 40
Enter number of epochs: 500
Results:
+-----------------+-------------------+---------------+--------------+
|  Architecture   |     Accuracy      | Training time | Testing time |
+=================+===================+===============+==============+
| [13, 40, 40, 1] | 98.40999999999997 | 36 m          | 36 m         |
+-----------------+-------------------+---------------+--------------+
```

Experiment 3(5 class)

Training set: 7000 points

Testing set: 7000 points

Features: 41



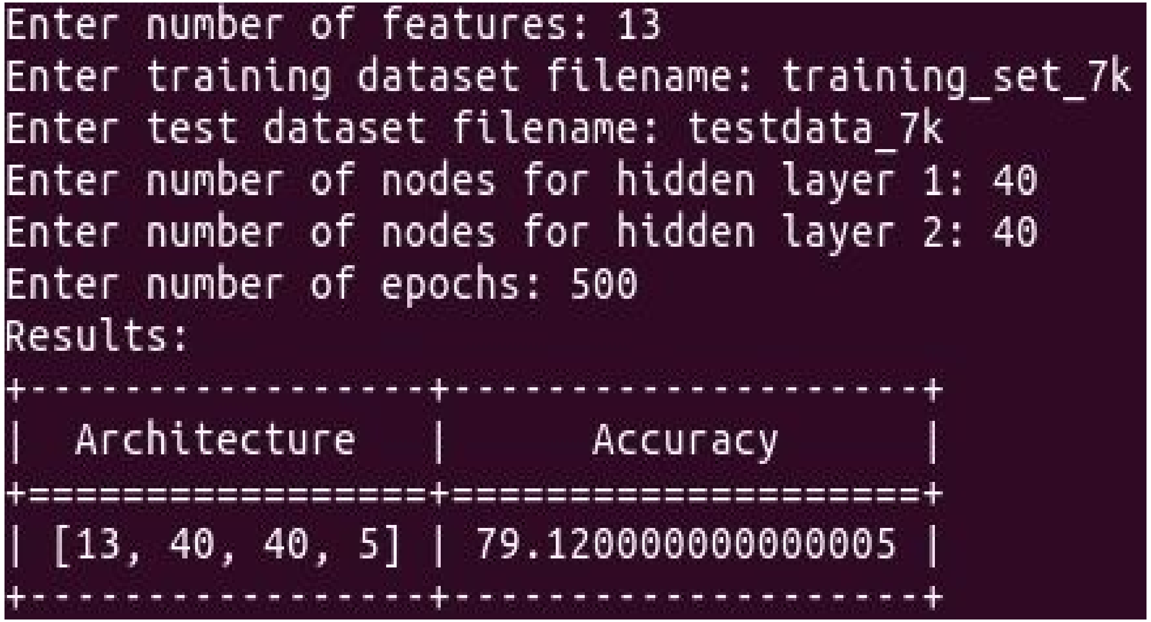
```
Enter number of features: 41
Enter training dataset filename: training_set_7k
Enter test dataset filename: testdata_7k
Enter number of nodes for hidden layer 1: 40
Enter number of nodes for hidden layer 2: 40
Enter number of epochs: 500
Results:
+------------------+---------------------+
|  Architecture    |      Accuracy       |
+==================+=====================+
| [41, 40, 40, 5]  | 80.34000000000003   |
+------------------+---------------------+
```

Experiment 4(5 class)

Training set: 7000 points

Testing set: 7000 points

Features: 13



```
Enter number of features: 13
Enter training dataset filename: training_set_7k
Enter test dataset filename: testdata_7k
Enter number of nodes for hidden layer 1: 40
Enter number of nodes for hidden layer 2: 40
Enter number of epochs: 500
Results:
+------------------+---------------------+
|  Architecture    |      Accuracy       |
+==================+=====================+
| [13, 40, 40, 5] | 79.12000000000005 |
+------------------+---------------------+
```