**Apoorva Vinod Gorur (gorur.2@wright.edu)**

**Mohammed Ibrahim Salman**

**|**

**Network attack detection using LSTM**

Table of Contents

[1 Introduction 3](#_Toc500882975)

[2 The Dataset 3](#_Toc500882976)

[2.1 Attack Types 4](#_Toc500882977)

[2.2 Data preprocessing 5](#_Toc500882978)

[2.2.1 Handling categorical Data and Normalization 5](#_Toc500882979)

[2.2.2 Dataset reduction 6](#_Toc500882980)

[2.3 Feature selection 7](#_Toc500882981)

[2.4 Class Distribution 8](#_Toc500882982)

[2.5 Stratified splitting 9](#_Toc500882983)

# 1 Introduction

In this digital age of the fast-growing internet where tons of terabytes of data are transferred every minute, there exists a growing need for a faster and suitable method to detect intrusions in computer networks. Various intrusion detection methods including signature-based systems and anomaly-based systems exist today and do an excellent job but there is still room for improvement. There is a need for more analysis and research about network security.

In our approach, we look at two models namely Long Short-Term Memory (LSTM) model and Multilayer Perceptron (MLP). We compare and analyze the performance of these two models with different configurations to determine which one is better suited for intrusion detection.

# 2 The Dataset

The KDD cup’99 [1] dataset is a modified version of the raw packet-level dataset issued by DARPA in 1998. MIT Lincoln lab processed and labeled the DARPA dataset. This is the dataset used for ‘The Third International Knowledge Discovery and Data Mining Tools Competition’. The task was to build an effective classifier that detected the ‘bad’ and ‘good’ connections. The raw packet-level data issued by DARPA has information related to each packet for connections between two hosts. The KDD dataset has connection level data that summarizes all the packets between two hosts into one connection. To illustrate this, figure 2 shows the packet level data for a conversation between two hosts in a network. Figure 3 shows a summarized version or connection level data of the same.

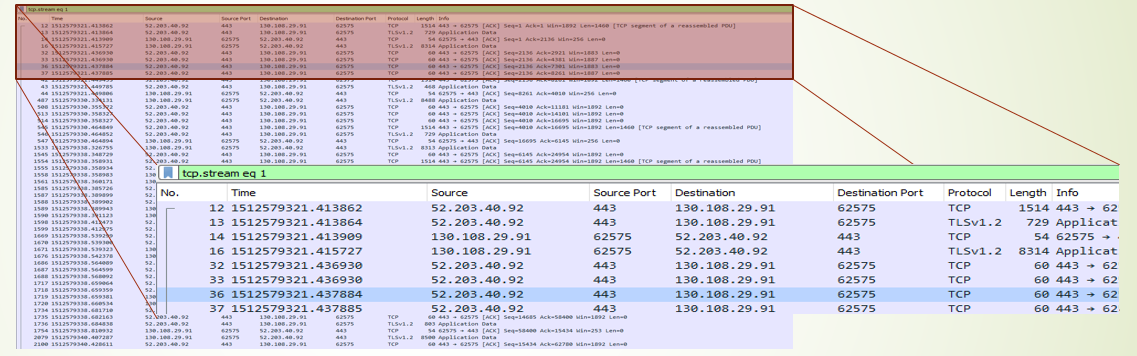


Figure 1 Packet level data



Figure 2 Connection level data

The original dataset consists of 4,898,431 data points. Each data point represents a session between two hosts in a network. Each vector has an attribute named ‘label.’ which denotes if it is normal or the type of attack if it is malicious. There is a total of 972781 data points labelled as normal and the rest 2952839 data points labelled as a type of attack.

## 2.1 Attack Types

The attacks are of 22 types which can be further grouped into 4 major types which are:

1. Denial of service(DoS): In this type of attack, the attacker tries to exhaust the target’s resources by spamming multiple requests. DoS comprises of the following attack types:

‘Neptune’, ‘back’, ’smurf’,’pod’, ’land’ and ‘teardrop’.

1. Probe: In this type of attack, the attacker tries to perform surveillance work for finding a vulnerability in the network. This type of attack includes ‘portsweep’, ‘satan’, ‘nmap’, ‘ipsweep’.
2. User to Root (U2R): In this type of attack, the attacker who already has access to a user’s account in the network tries to access the root and perform malicious activities. This includes the attacks ‘buffer\_overflow’, ‘loadmodule’, ‘rootkit’ and ‘perl’.
3. Remote to Local (R2L): In this type of attack, the attacker tries to gain access to a user account in the network in order to access the root. This includes the attacks 'warezclient', ' multihop', ' ftp\_write', 'imap', 'guess\_passwd', 'warezmaster', 'spy' and 'phf

The table 1 below shows the number of records for each type of attack,

|  |  |  |
| --- | --- | --- |
| Class of attack | Attack | count |
| DoS | neptune | 1072017 |
| back | 2203 |
| teardrop | 979 |
| pod | 264 |
| land | 21 |
| Probe | satan | 15892 |
| ipsweep | 12481 |
| portsweep | 10413 |
| nmap | 2316 |
| R2L | warezclient | 1020 |
| guess\_passwd | 53 |
| warezmaster | 20 |
| imap | 12 |
| ftp\_write | 8 |
| multihop | 7 |
| phf | 4 |
| spy | 2 |
| U2R | buffer\_overflow | 30 |
| rootkit | 10 |
| loadmodule | 9 |
| perl | 3 |

Table 1 Classes of the attacks and their counts

Table 2 shows the counts of the attacks after grouping everything into 4 categories.

|  |  |
| --- | --- |
| Class of attack | count |
| DoS | 3883370 |
| Probe | 41102 |
| R2L | 1126 |
| U2R | 22 |

Table 2 Attacks after grouping

As seen from the table 2 the dataset is unbalanced for the classes ‘R2L’ and ‘U2R’. This imbalance may lead the classifier to be more biased towards the high instance classes and prevent the learning of classes which are imbalanced.

## 2.2 Data preprocessing

The following subsections give information about data preprocessing and what steps were taken during feature selection.

### 2.2.1 Handling categorical Data and Normalization

4 fields in the dataset are categorical. They are ‘protocol\_type’ , ‘service’, ‘flag’ and ‘label’.

* Protocol\_type: ‘icmp’(2833545), ‘tcp’(1870598), ‘udp’(194288)
* Service: 70 types in total
* Flag: 11 types in total
* Label: 23 types. 22 attacks and 1 label for normal connections.

‘protocol\_type’ and ‘flag’ have been transformed with dummy variables (OneHotEncode). ‘service’ column has been label encoded which means all the types in the ‘service’ attribute is associated with a number from 1-70.

The 22 types of attacks have been further grouped into 4 types of attacks:

* DoS (Denial of service)
* Probe
* R2L
* U2R

Now the ‘label’ attribute has 5 groups; 4 for attacks and 1 for normal connections. This attribute is then transformed with dummy variables(OneHotEncode). The resulting data after the above transformations now have only numerical data. The data is normalized to the range 0-1 using the MinMaxScaler [2] from sci-kit learn.

### 2.2.2 Dataset reduction

One of the deficiencies of the KDD dataset is that there is a substantial number of redundant records. This leads the classifier to be more biased towards the records that are abundant and leads to the classifier not learning the records that are very few like the U2R and R2L attacks. Another reason for dataset reduction is due to hardware limitations. The classification of the whole dataset using any of the models takes more than 3 days on a computer with specifications described in section 4. Figure 4 shows the distribution of the whole dataset and figure 5 shows the distribution of the whole dataset. As figure 5 shows, the dataset’s reduction is adjusted to match the distribution of the original dataset.

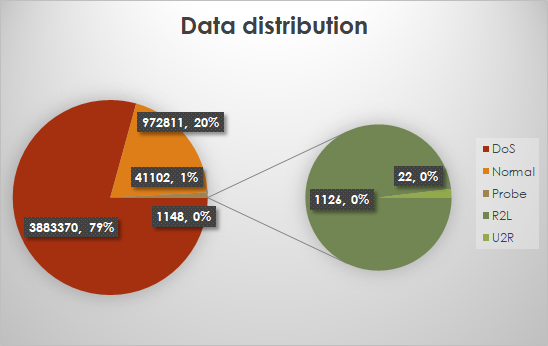


Figure 3 Data distribution before data reduction

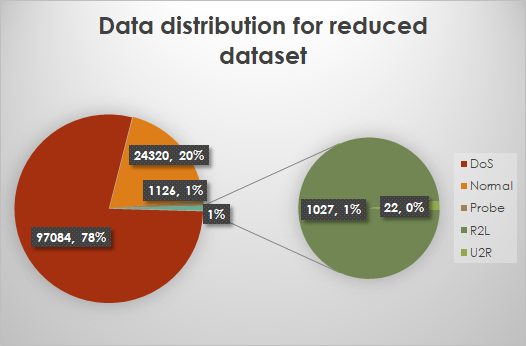


Figure 4 Data distribution after data reduction

## 2.3 Feature selection

Figure 6 shows the feature importance graph for the features in our dataset. The features ‘logged\_in’, ‘count’, ‘srv\_count’, ‘same\_srv\_rate’, ‘protocol\_type\_icmp’, ‘dst\_host\_same\_src\_port\_rate’, ‘dst\_host\_count’, ‘flag\_SF’, ‘dst\_host\_same\_srv\_rate’, ‘protocol\_type\_udp’ and ‘protocol\_type\_tcp’ are shown to be the top ten features that influence the output. Based on this information, the n\_components for pca is chosen as 10.

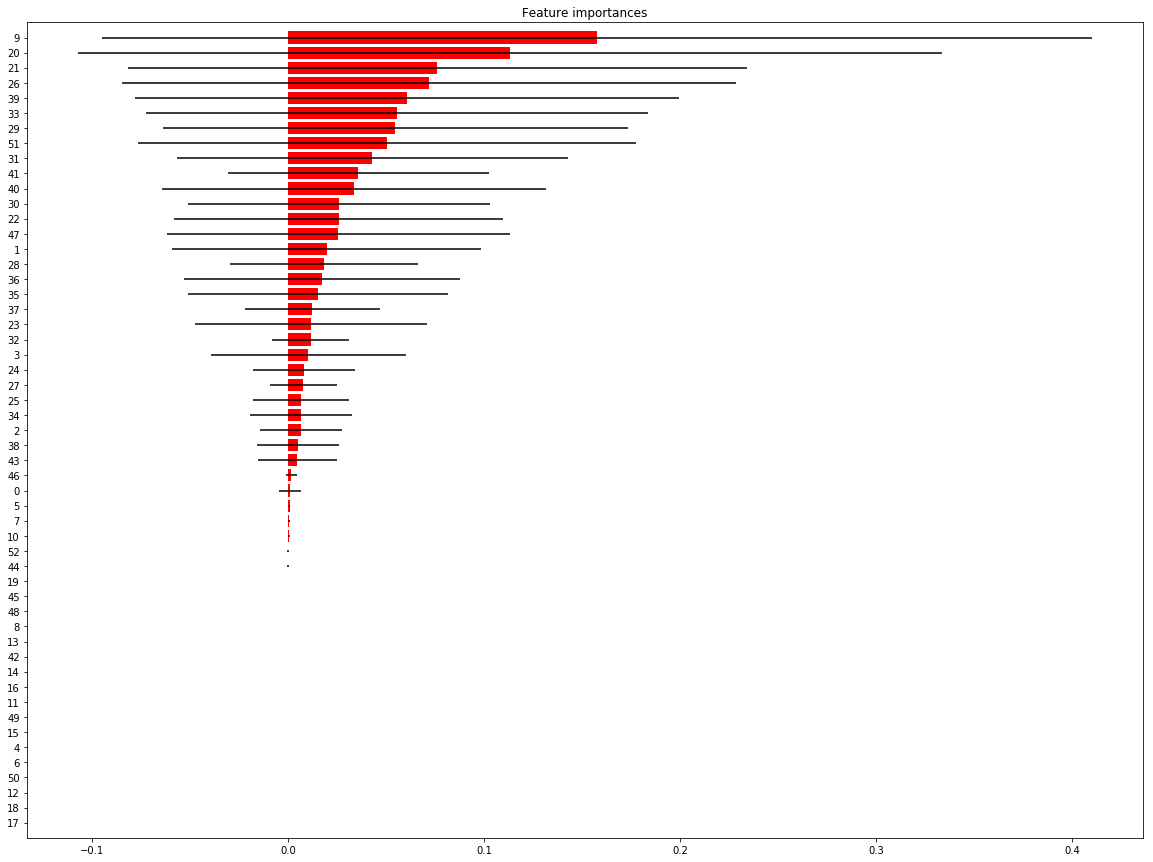


Figure 6 Feature importance graph

## 2.4 Class Distribution

Figure 7 shows the LDA (Linear discriminant analysis) of the dataset with 2 components. The figure shows significant overlapping between the classes. The class ‘U2R’ overlaps with ‘R2L’ and ‘Normal’ as well. As we see in section 4.2, this causes some of the ‘U2R’ classes to be misclassified as ‘Normal’ most of the time.

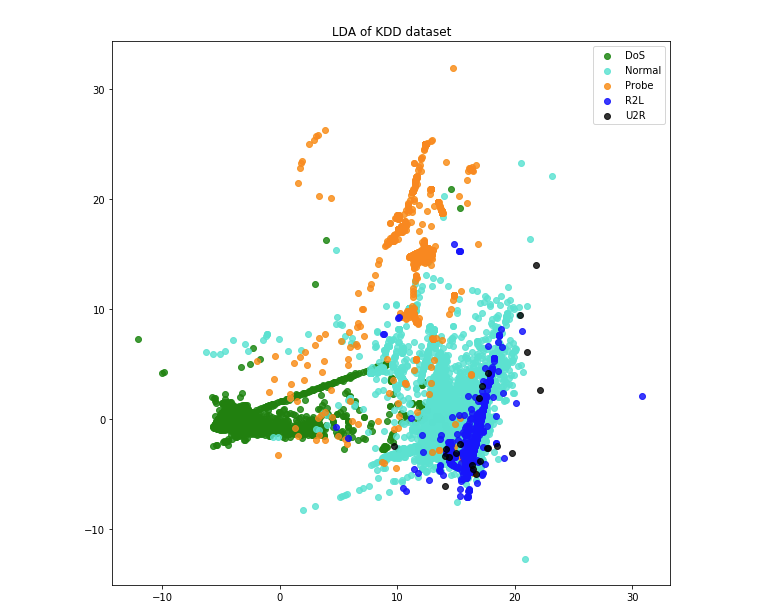


Figure 6 LDA of the dataset

## 2.5 Stratified splitting

The dataset is divided into training, validation and testing sets in the ratio 60:20:20. Since the dataset is very big, even when it is reduced by the ratio, there are enough samples for training, testing, and validation. The splitting is done using sci-kit learn’s train\_test\_split [3] library with the stratification option turned on. Stratified splitting distributes the dataset with an equal number of proportions for each class that is specified.

# 3 Experiment

The models will be run with different configurations with layers varying from 1-4 while number of nodes is set to 15,50 and 100. Results and analysis will be included in a separate document. Note that the training takes a long time, hence this section might take a while to be completed.