A novel clustering-based autoencoder driven intelligent intrusion detection approach

Team-6 Members

- Alimurtaza Mustafa Merchant - 191IT205
- Naveen Shenoy - 191IT134
- Sohanraj R - 191IT149
- Suyash Chintawar - 191IT109

Introduction

- Network intrusion attacks have become more complex and increasingly difficult to detect and can cause significant economic and social consequences.
- ML techniques in the intrusion detection task suffer from latency and computational complexity issues, and can be ineffective in learning complex and non-linear relationships that exist within big datasets.
- NSL-KDD dataset is publicly available dataset by Canadian Institute for Cybersecurity
- We propose a novel clustering based deep learning model for Intrusion Detection.

Problem Statement

To build an autoencoder based deep learning classifier supported by clustering techniques for performance improvement for the task of intrusion detection.

Dataset

NSL-KDD Dataset

- The NSL-KDD dataset has a total of 41 features which includes 38 numeric features and 3 symbolic ones.
- Numeric features: source bytes, destination bytes, num failed logins, wrong fragment, etc.
- Symbolic Features: protocol type, service & flag.
- A total of 4 types of attacks are labeled:- DoS, R2L,U2R and Probe.
- Training data: 125,973 instances, Test data: 18,793 instances

Dataset

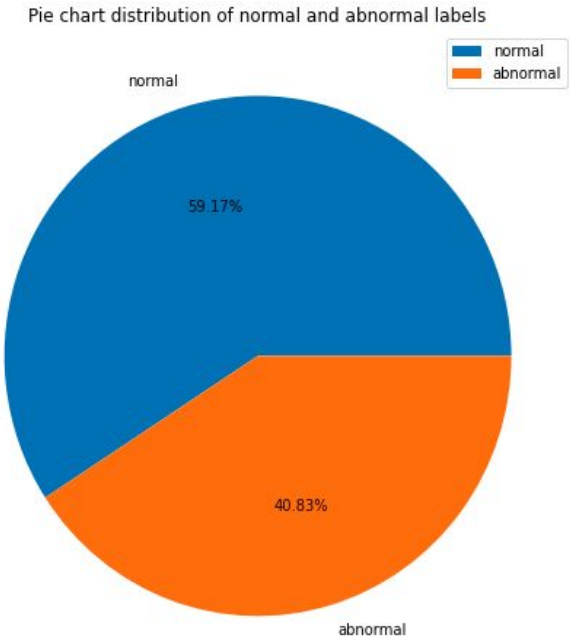
No.	Features	Types	No	Features	Types
z_1	duration	cont	z_{22}	is guest login	cont
z_2	protocol type	symb	z_{23}	count	cont
z_3	service	symb	z_{24}	srv count	cont
z_4	flag	symb	z_{25}	serror rate	cont
z_5	source bytes	cont	z_{26}	srv serror rate	cont
z_6	destination bytes	cont	z_{27}	rerror rate	cont
z_7	land	cont	z_{28}	srv rerror rate	cont
z_8	wrong fragment	cont	z_{29}	same srv rate	cont
z_9	urgent	cont	z_{30}	diff srv rate	cont
z_{10}	hot	cont	z_{31}	srv diff host rate	cont
z_{11}	num failed logins	cont	z_{32}	dst host count	cont
z_{12}	logged in	cont	z_{33}	dst host srv count	cont
z_{13}	num compromised	cont	z_{34}	dst host same srv rate	cont
z_{14}	root shell	cont	z_{35}	dst host diff srv rate	cont
z_{15}	su attempted	cont	z_{36}	dst host same src port rate	cont
z_{16}	num root	cont	z_{37}	dst host srv diff host rate	cont
z_{17}	num file creations	cont	z_{38}	dst host serror rate	cont
z_{18}	num shells	cont	z_{39}	dst host srv serror rate	cont
z_{19}	num access files	cont	z_{40}	dst host rerror rate	cont
z_{20}	Num outbound cmds	cont	z_{41}	dst-host srv rerror rate	cont
z_{21}	is host login	cont			

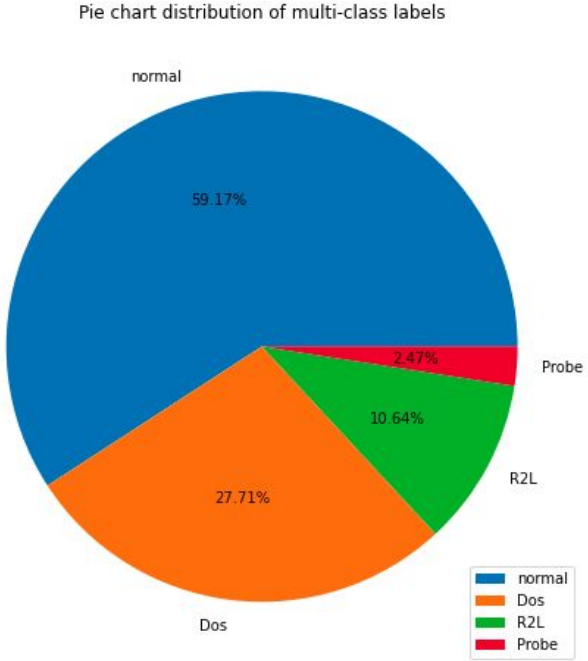
Fig: Features of NSL-KDD dataset

Attack class	Attack profile
DoS	back, neptune, land, smurf, pod, teardrop
R2L	ftp write, imap, guess passwd, multihop, phf, warezclient, spy, warezmaster
U2R	loadmodule, buffer overflow, rootkit, perl
Probe	nmap, ipsweep, satan, portsweep

Fig: Attack profiles of different attack classes

Data Distribution





Methodology

Low level preprocessing

- Firstly, as the raw dataset does not contain column names, we assign respective column names as given in the description of the official dataset.
- Furthermore, the attack labels for each sample are converted to their respective attack classes namely, DoS, U2R, Probe, and R2L.
- Finally, the dataset is also parsed to check for any missing values or null data.

Methodology

Outlier Analysis

- Outliers are known to interfere with the learning process causing miss detection in intrusion detection systems.
- To identify them, we have used the Median Absolute Deviation Estimator (MADE) as follows:

$$MADE = P * med(z_{fj} - |med(z_{fj})|)$$

$$z_{fj} > p * MADE$$

where P=1.4826 represents a multiplicative constant typically used under the assumption of data normality and p=10.

- It is worth mentioning that the dataset is significantly unbalanced, including very less test instances in the U2R attack category. Hence, the U2R class was removed from the final train and test datasets.

Methodology

Normalization

Min-Max scaling to normalize data between -1 to 1

$$X_{sc} = (X - X_{min})/(X_{max} - X_{min})$$

One-hot encoding of features:

- The 3 categorical features ***protocol type, service, flag*** were transformed into numerical values using one-hot-encoding. For example, the ***service*** feature has three attributes: tcp, udp and icmp.
- Overall, the 41-dimensional features were mapped into 118-dimensional features (38 continuous and 80 with binary values related to the features z2, z3, z4).

Methodology

Feature Extraction

- We use Pearson Correlation Coefficient (PCC) to find the most correlated numeric features and only the features with correlation > 0.5 were selected.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

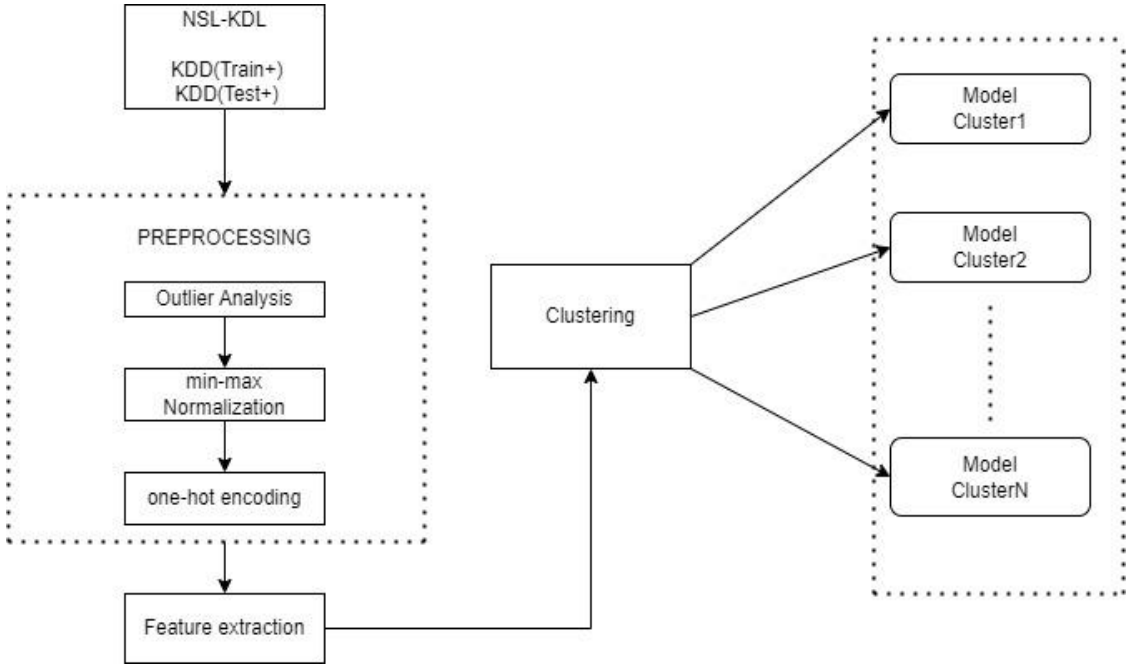
- The selected numerical features were then combined with the one hot encoded categorical features to obtain the final dataset.
- The same feature extraction technique was used to obtain the datasets for binary as well as multi-class classification.

Methodology

Classification using Clustering

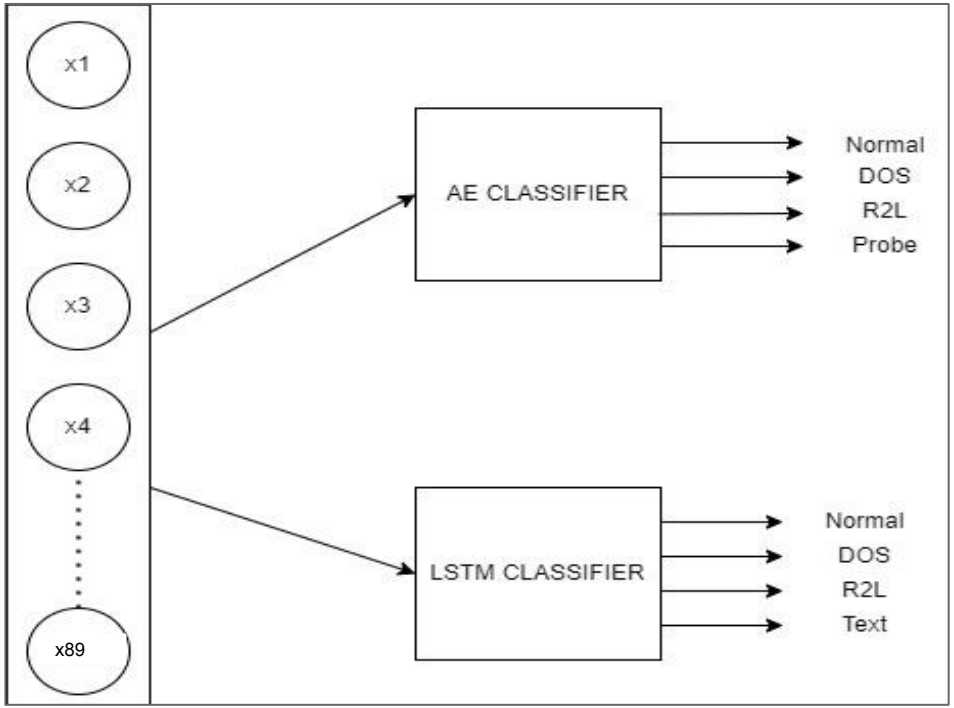
- Clustering of the data samples is performed to identify similar data points. K-means clustering has been used for this purpose.
- The LSTM/autoencoder classifiers are then trained independently on each of the formed clusters.
- To obtain predictions on the test data, each test data point is assigned to one of the clusters formed by the training data. The predictions of the test data points are then obtained using the predictions of the respective cluster.
- In the combined model, for each cluster we assign the model (autoencoder/lstm) which performs better on that cluster.

Methodology



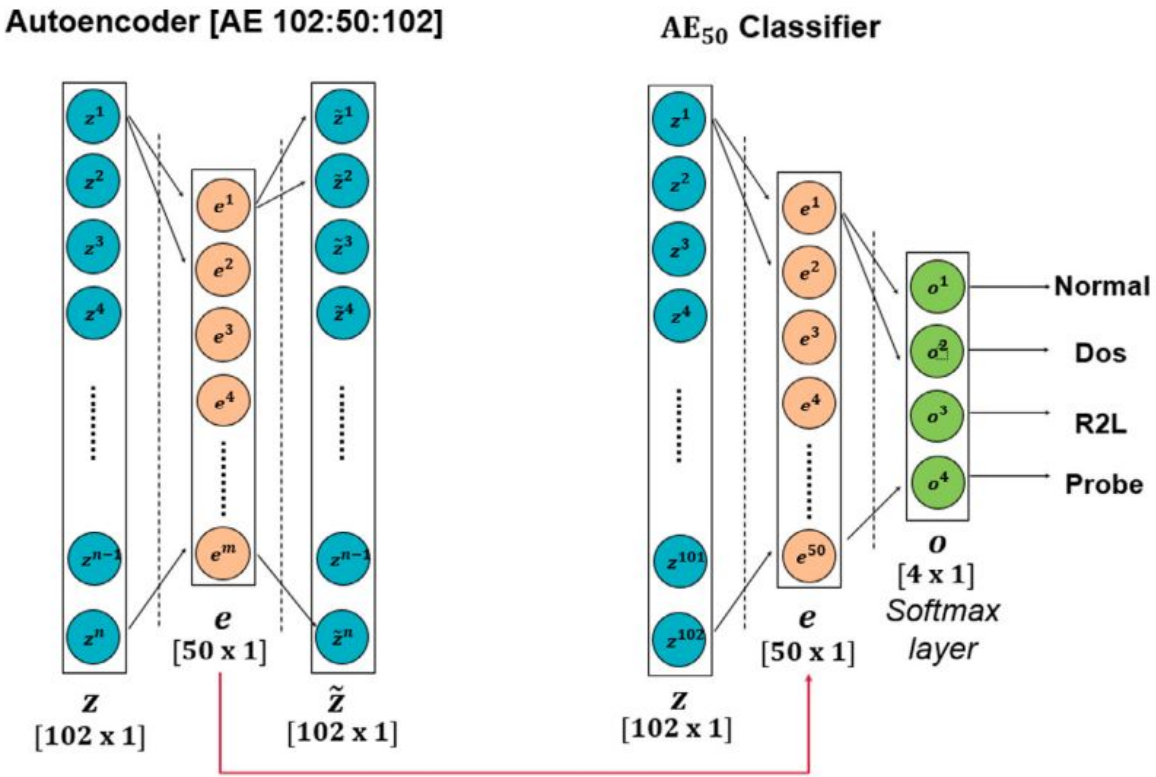
The figure alongside shows part-1 of the methodology which consists of forming clusters on which models will be trained independently

Methodology



The figure alongside shows the two types of classifier models to be applied for prediction of classes.

Autoencoder Architecture



Results

Model	Accuracy (%)
AE_{50}	82.75
$AE_{64,50}$	81.50
AE_{64}	80.30
$AE_{64,32}$	80.74

Experimentation with different Autoencoder architectures for the binary classification task

Results

Model	Accuracy	Precision	Recall
LSTM	77.73	78.01	77.43
Auto-Encoder	80.22	79.23	79.21
Clustered LSTM	76.29	79.68	76.29
Clustered Auto-Encoder	82.40	80.55	82.40
Clustered combined	**83.06**	**82.04**	**83.06**

Multi-class Classification

Model	Accuracy	Precision	Recall
LSTM	80.68	71.07	87.64
Auto-Encoder	82.75	72.88	87.39
Clustered LSTM	81.25	71.76	**88.03**
Clustered Auto-Encoder	89.32	86.03	87.68
Clustered combined	**90.17**	**90.07**	84.93

Binary Classification

Conclusion

- This research attempts to tackle the problem of intrusion detection by using a novel clustering based classification approach.
- K-means clustering was used to cluster the training samples into multiple clusters and deep learning models like LSTM and Autoencoders were employed for the task of classification.
- It was observed that the proposed approach performed comparable to the state-of-the-art techniques, rather, outperformed in some of the scenarios.
- The autoencoders performed better than the LSTM classifiers for this particular task. Also, the combined model of LSTM and autoencoders outperformed each of the models when used independently.

THANK YOU