

B23BH01



Intrusion Detection System Using AIML

Guide: Dr. Bheemappa Halavar

Team Members



Mohd Hamza
S202000101



Lakshya Boob
S20200010109



Nasam Venu
S20200010225

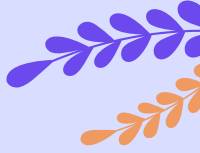
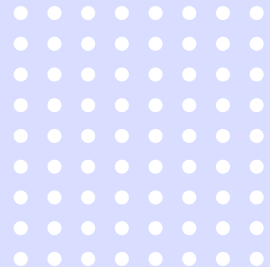


TABLE OF CONTENTS

- 1. Introduction**
- 2. Problem Definition & Motivation**
- 3. Literature Survey**
- 4. Proposed Methodology**
- 5. Experiment**
- 6. Results**



Timeline

BTP-1

We have studied some research paper and tried to figure out the problem statement and algorithm used,

Did the literature review Hybrid Feature Selection, Balancing the dataset(Just SMOTE) , Basic ML model (RF and DT) evaluation

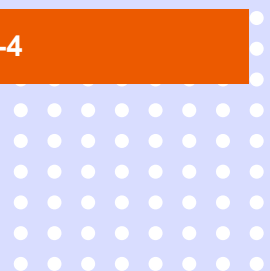
BTP-2

BTP-3

Novel approach to improve the imbalance ratio of dataset , Ensemble Model , Evaluation Metrics and Literature Review

Optimization and concluding the research

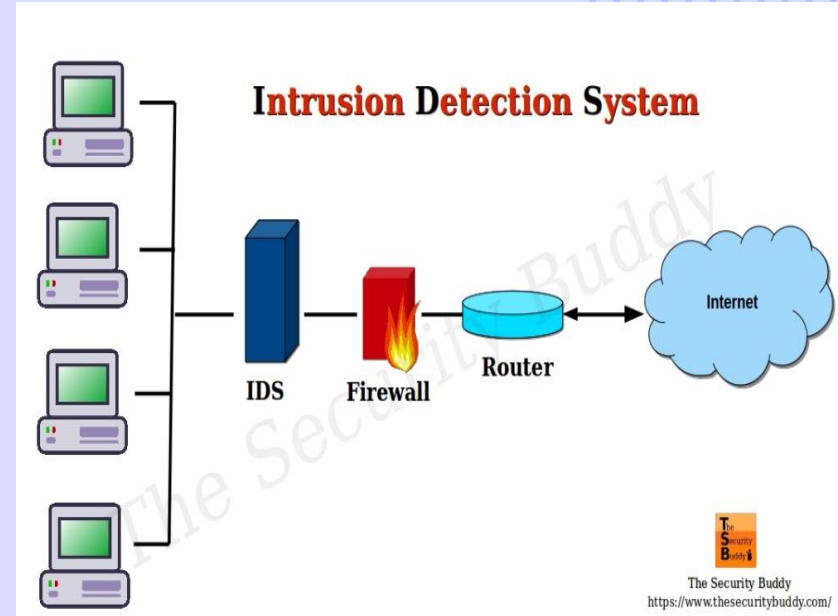
BTP-4



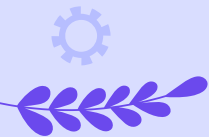
Intrusion Detection System



- 👉 security technology designed to monitor computer systems and networks for signs of unauthorized access, malicious activity.
- 👉 IDSs work by analyzing network traffic and system activity to detect patterns or anomalies that may indicate an intrusion or security threat.
- 👉 When an IDS identifies suspicious activity, it generates an alert to notify security personnel or automated systems, allowing them to respond quickly and prevent potential damage.



Misra, Amrita. *What Is IDS or Intrusion Detection System and How Does It Work?*
<https://www.thesecuritybuddy.com/data-breaches-prevention/what-is-ids-intrusion-detection-system-how-does-it-work/>.




02 & 03

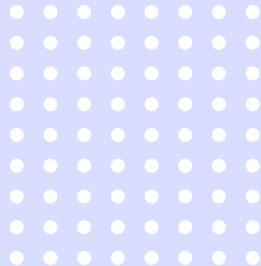
Problem Definition & Motivation



Problem Definition



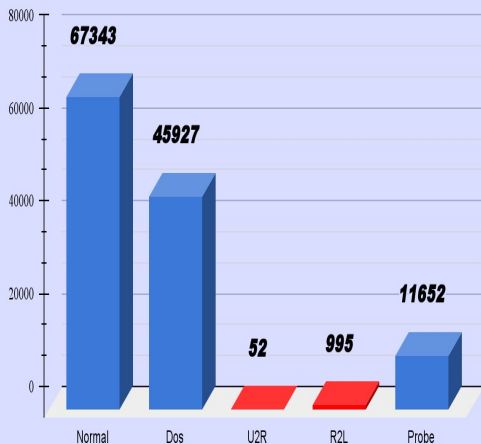
To design an IDS system with better performance metrics in real time scenario using ML algorithms on imbalanced dataset and to reduce the computational power used in training and testing.



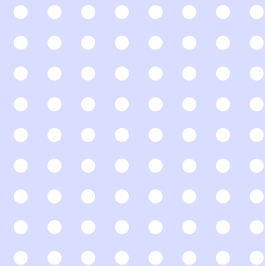
Motivation

- 👉 Performance of IDS can be improved.
- 👉 Imbalanced datasets can lead to bias in the IDS, where it becomes more accurate at detecting the majority class but less effective at identifying the minority class.
- 👉 Moreover, training the model consumes lot of computational power.

NSL-KDD Training Dataset



Datasets

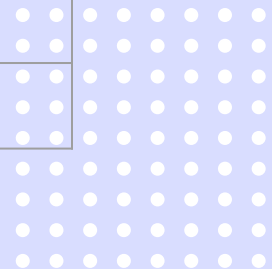
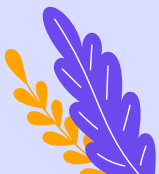


Issue with the Datasets

Imbalance Ratio

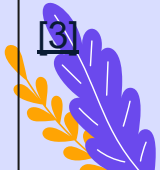
- It is defined as the fraction between the number of instances of the majority class and the minority class.

Dataset	Imbalance Ratio
NSL-KDD	648
CSE-CIC-IDS2018	53,887
CIC-IDS2017	112,287
KDD Cup99	36,725



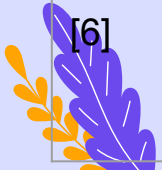
Literature Survey

Reference	Dataset	Algorithms	Results	Merits	Scope
[1]	NSL-KDD	IDS-SMOTE-RF	Accuracy- 99.89 Precision -99.87 Recall- 99.88 F1 score- 99.87	The metrics are good..	1. Recent dataset can be used. 2.Computation time reduction
[2]	CSE-CIC-IDS 2018	Spearman rank correlation. Ensemble learning (lr,dt and gb)	Accuracy- 98.8 Precision -98.8 Recall- 97.1 F1 score- 97.9	Accuracy is decent keeping in mind the fact that detection time was reduced by 1/3rd. For ensemble learning best 3 classifiers were selected by comparing 7 ml classifiers.	1. Imbalance ratio of dataset should be lowered. 2. DL algorithms should be considered. 3.While comparing the classifiers hyperparameter technique should be used for tuning them.
[3]	NSL-KDD	Hybrid feature selection approach (chi square,anova,pca) DNN classifier	Accuracy- 99.73 Precision -99.75 Recall- 99.73 F1 score- 99.72 Time-2.7 seconds	Less training and testing time due to the use of hybrid feature selection approach. Evaluation metrics are good.	1. Recent dataset should be used. 2.Multiclass classification can be done. 3. Training data should be balanced.



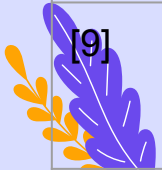
Literature Survey

Reference	Dataset	Algorithms	Results	Merits	Scope
[4]	CSE-CIC-IDS2018	SMOTE adaboost,DT,RF,KNN, Gradient Boosting	Accuracy ADA-99.60 DT-99.57 RF-99.3 KNN-98.58 GB-99.29	1. Imbalance ratio is reduced by 1000 times. 2.Accuracy is great with multiclass classification. 3. Various ml classifiers are compared	1. Time can be reduced by employing various techniques. 2. DL algorithms should also be considered.
[5]	CICIDS2017	IDS-SMOTE Random Forest,Naive Bayes,k-NN SVM,Decision jungle and decision forest	SVM emerged as the best performer	1. Addresses class imbalance problem. 2. It Evaluates different SMOTE Variants 3. Better Experiment Evaluations.	1. DL algorithms should be considered. 2. Recent Dataset can be used.
[6]	NSL-KDD	PCA with Random Forest,svm ,Decision Tree	Performance time (min) :3.4min Accuracy:96.78 Error Rate:0.21	1.the performance time is greatly reduced.(Like DT take 12min) 2.The pca is used for the reduction of the dimension.	1.Different Features selection method can be used. 2.Multiclass classification can be done.



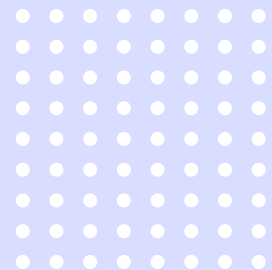
Literature Survey

Reference	Dataset	Algorithms	Results	Merits	Scope
[7]	NSL-KDD	GAN, Knn,DT,SVM,ANN	Accuracy KNN-91.60 DT-91.9 RF-91.45 SVM-90.38 ANN-90.93	1. Imbalance ratio is reduced by using GAN (adversarial traffic).. 2.Accuracy is great with multiclass classification. 3. Various ml and dl classifiers are compared	1. Time can be reduced by employing various techniques. 2. Accuracy can be increased. 3. GAN is originally made for data augmentation in images and not oversampling
[8]	NSL-KDD	ANOVA, NB,DT,RF,GBDT, SVM,KNN,ANN GAN for data generation	Accuracy: NB-72.39, DT-97.7, RF-99.84, GBDT-99.62, KNN-98.19, ANN-96.67	1. Addresses class imbalance problem. 2. Really good evaluation metrics.	1. No Multiclass classification. 2. Recent Dataset can be used.
[9]	CIC-IDS 2017	GAN Based oversampling, CNN	Accuracy- 93.74	1. Addresses class imbalance using gan 2. Multiclass classification	1. GAN is originally made for data augmentation in images and not oversampling



Existing Work

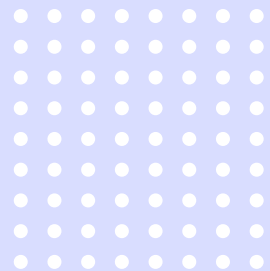
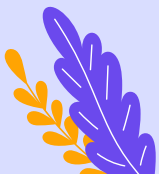
- NSL-KDD dataset.
- Employed hybrid feature selection method to reduce the complexity of the model and improve its metrics.
- Used SMOTE to balance the dataset.
- Used GAN separately
- Used SMOTE and GAN together.
- Analysis of various ML algorithms and reported their metrics.
- Created an ensemble model based on the results
- The metrics were comparable with existing literature.



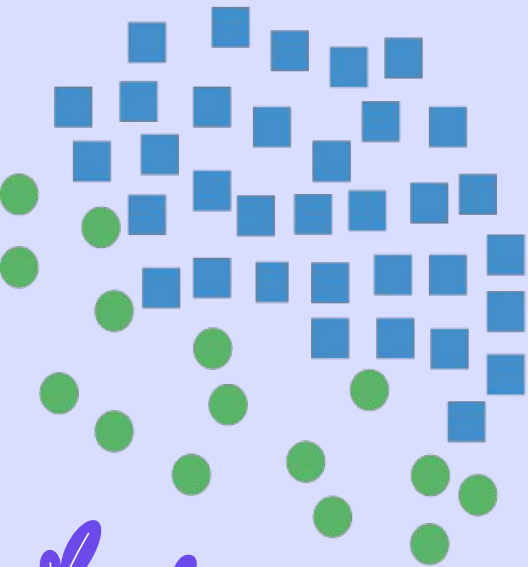
Algorithms Used

SMOTE(Synthetic minority oversampling technique)

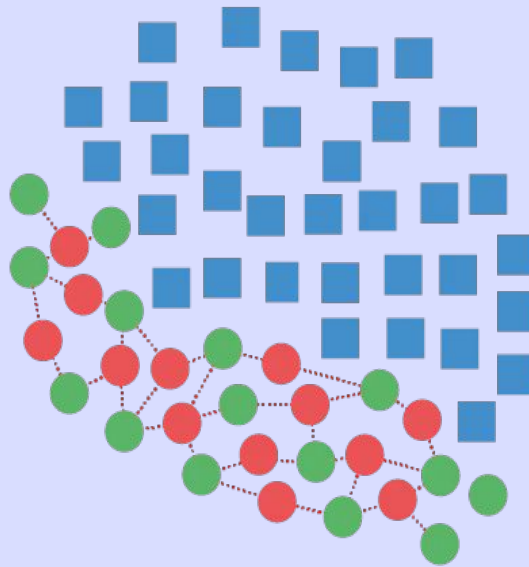
- Uses KNN approach
- K nearest neighbours to a sample of minority class are selected randomly.
- New samples are synthetically generated based on the linear combination of the two.
- $x_{ni} = \min(x_{ji}, x_{ki}) + |x_{ji} - x_{ki}| \times r$; x_{ni} : ith feature of nth new sample,
r: random number between 0 and 1,
 x_{ji}, x_{ki} : random 2 samples from minority class



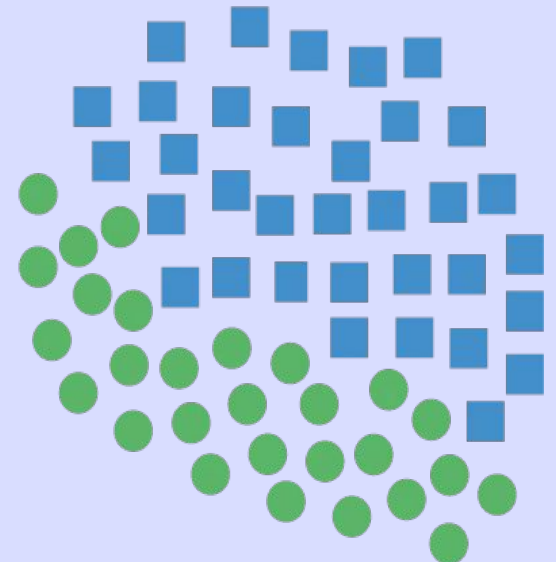
Synthetic Minority Oversampling Technique



Original Dataset



Generating Samples

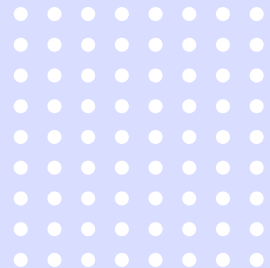


Resampled Dataset

Algorithms Used

Drawbacks of SMOTE

- It can't generate diverse samples as new samples will only be generated in between the old samples.
- This can be an issue in datasets which are not well clustered.
- On the contrary, it may also oversample the noisy samples at the same time.



Algorithms Used

GAN (Generative Adversarial Network)

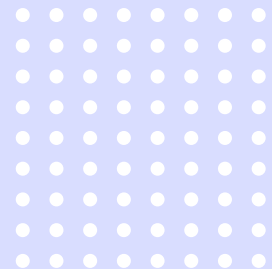
- Originally designed to generate realistic looking 'fake' images.
- Can also be modified to generate fake minority class samples.
- Consists of 2 neural networks which compete against each other.
- Generator - generate data similar to real data using random values.
- Discriminator - takes real data and fake data from generator and tries to classify them correctly



Algorithms Used

Drawbacks of GAN

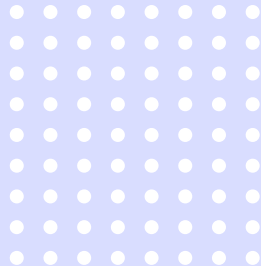
- It isn't originally designed for oversampling but for data augmentation.
- It can be ineffective sometimes due to random noise.
- It can in turn face data scarcity problem as minority class samples are less.



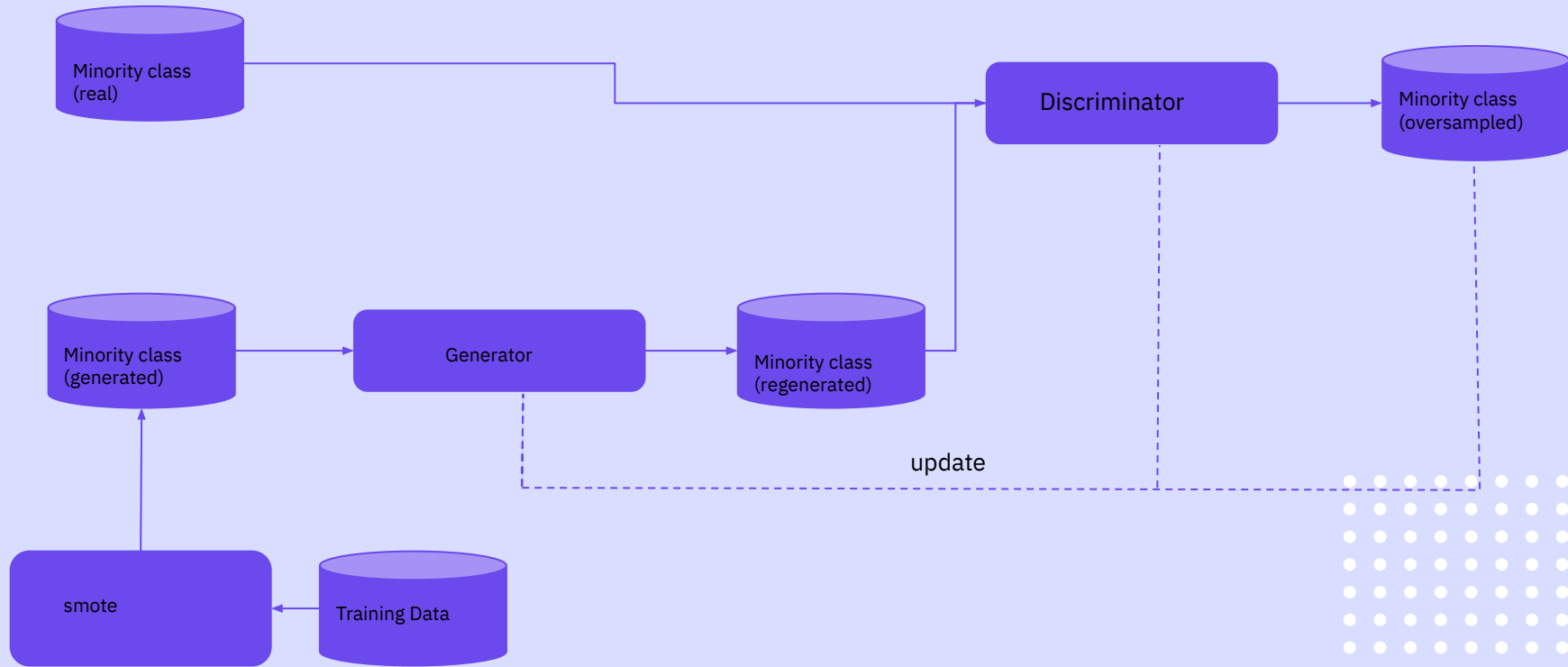
Algorithms Used

SMGAN- Novel approach

- SMOTE and GAN is used together exhibiting 'transfer learning'.
- Instead of random samples , generator uses oversampled data from SMOTE to begin with.



SMGAN



Algorithms Used

SMGAN- Novel approach

- Generator Score

$$\min_G - E_u [\log D(G(u))]$$

- Discriminator Score

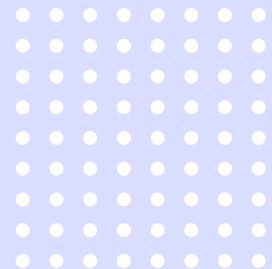
$$\max_D E_{x^*} [\log D(x^*/x)] + E_x [\log (1 - D(G(u)))]$$

x^* : Training samples of minority class

u : Oversampled data generated from smote

$D(x^*/x)$: probability that x^* is real data

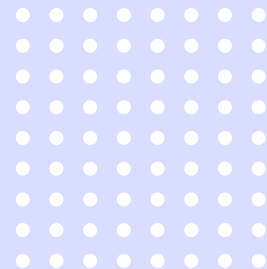
$D(G(u))$: probability that data from generator is real



Algorithms Used

Pseudo Code

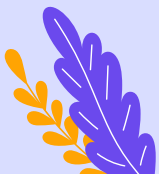
```
u ← call SMOTE(x*, k) // x* : minority samples  
u ← call GAN(x*, u, N-n)
```



Algorithms Used

Pseudo Code (for gan)

```
// gd: gradient for discriminator
// wd: weights for discriminator
// gg: gradient for generator
// wg: weight for generator
// mi is minibatch
// u: oversampled data from smote
while(epochs--){
    gd ← SGD(- log(D(x)) - log(1 - D(G(u))), wd , mi)
    wd ← weights(gd,wd)
    gg ← SGD(-log(D(G(u))), wg, mi)
    wg ← weights(gg,wg)
}
newdata = generator(u)
```

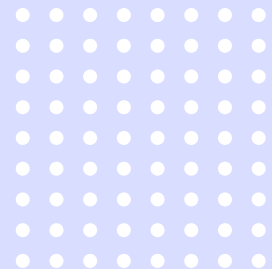


Algorithms Used

Pearson Correlation

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

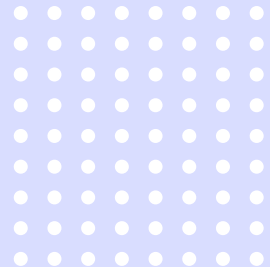
- x_i, y_i : sample points , \bar{x}, \bar{y} : mean
- $r_{xy} > 0$: positive correlation (redundant features)
- $r_{xy} = 0$: no correlation
- $r_{xy} < 0$: negative correlation



Algorithms Used

Principal Component Analysis

- Reduces the dimension of data.
- $y = A(X - U)$
 - y : projected point on the new subspace
 - x : original point
 - u : Mean vector
 - A : Top r eigenvectors of covariance matrix of x



Algorithms Used

Decision Tree

- Impurity in the dataset is tried to reduce at every decision.
- This is determined by Entropy

$$\text{Entropy}(S) = - \sum_{c \in C} p(c) \log_2 p(c)$$



Algorithms Used

Random Forest

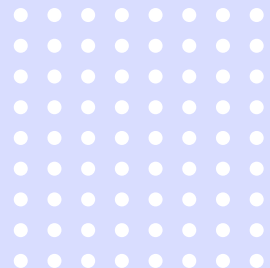
- More than one decision tree are used to predict the outcome.

KNN

- Prediction is made based on k nearest neighbors.
- Euclid distance is used to find nearest neighbors.

SVM

- Data is mapped to higher dimensions.
- Kernel Trick is used for implicit mapping.



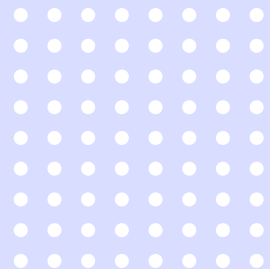
Algorithms Used

Extra Tree

- Similar to Random Forest in approach.
- Computationally faster than RF

MLP

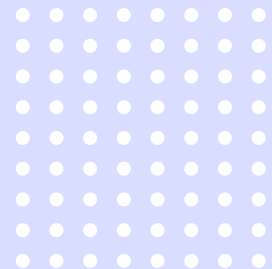
- Multi Layer Perceptron.
- Can represent any function with enough number of hidden units.



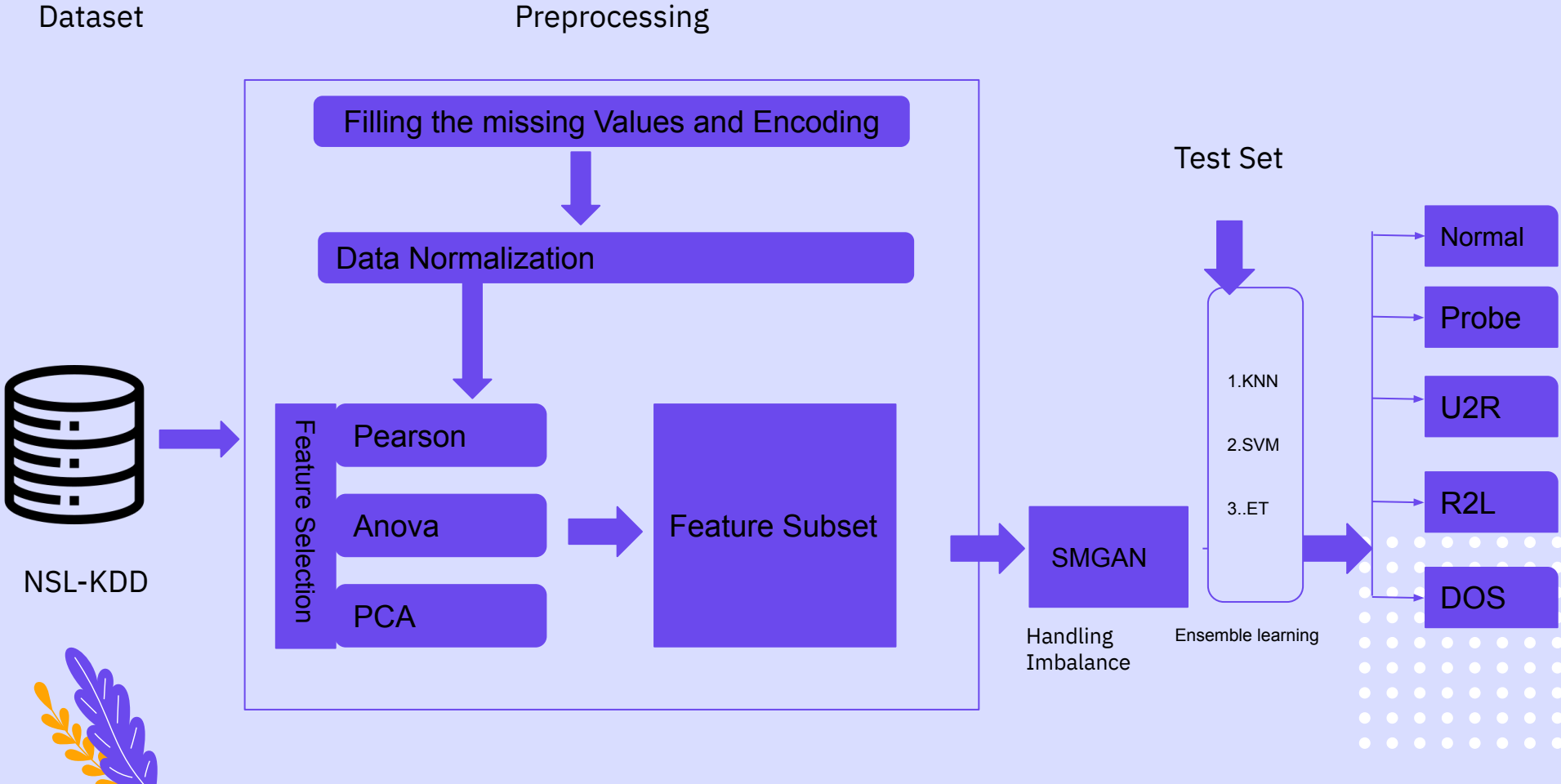
Algorithms Used

Ensembling

- Combining various classifiers can increase the overall accuracy.
- This is done using various techniques
 - Voting
 - Bagging
 - Boosting
 - Stacking



Proposed Methodology



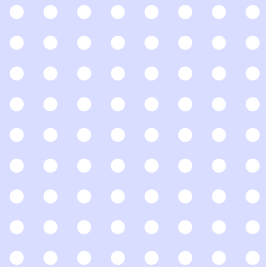
06 & 07

Experiment & Result



Data Preprocessing

- Missing values were filled with mean of their respective column.
- Encoding was done on 'protocol_type', 'service' and 'flag' features along with the class variable.
- 'Num_outbound_cmds' feature had all zero values and so was removed.
- Normalization was done to scale the values between 0 and 1.

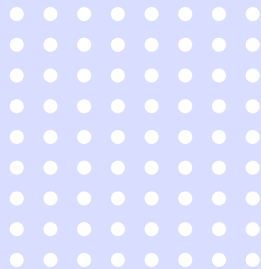


Feature Selection

Dataset was splitted into training set and test set in the ratio 3:1.

Feature Selection

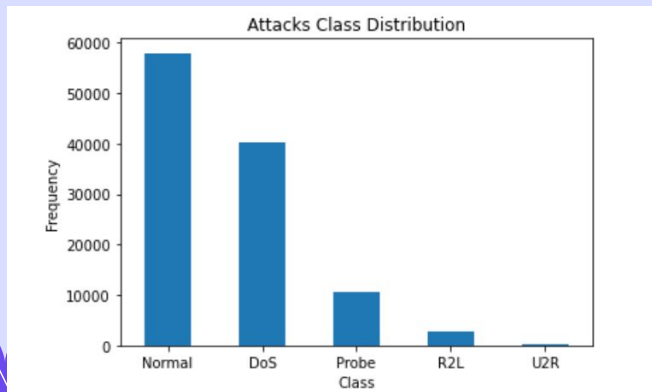
- Hybrid method involving Anova, Pearson correlation and PCA was used to form a feature subset.
- Input Data is reduced to 56% (23 out of 41 features).
- Removed features are considered noisy and irrelevant.



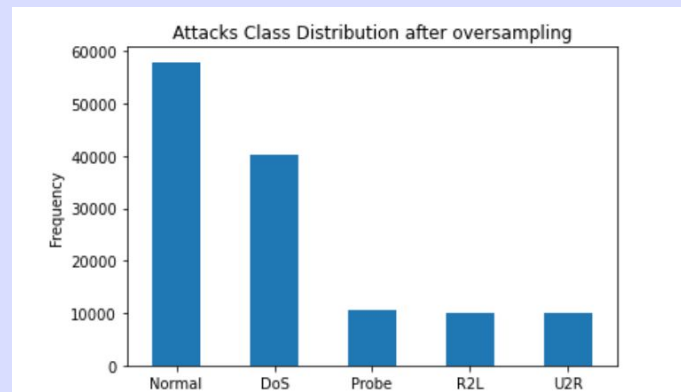
Handling Imbalance

Handling Imbalanced Dataset

- SMGAN Algorithm was used on training set.
- Imbalance ratio was reduced to 5.7965 from 648.
- Samples of 'U2R' and 'R2L' classes were increased from 85 to 10000 and 2766 to 10000 respectively.



Before



After

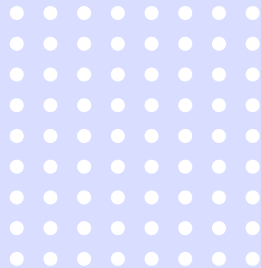
Handling Imbalance

Implementation details of SMGAN

- $k=4$ for SMOTE.
- Generator Neural Network
 - a. 3 hidden layers with Relu activation
 - b. Sigmoid function in last layer.
 - C. input and output dimension - 23
- Discriminator Neural Network
 - a. 2 hidden layers with leaky relu
- Loss function - BCE

Before

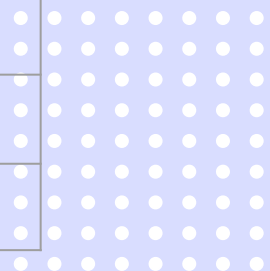
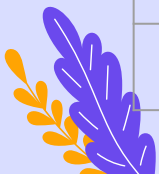
After



Classification

- Initially Various classifiers were trained separately:

Algorithm	Hyperparameters
SVM	Kernel = 'rbf' , C=1, Gamma=10
KNN	K = 5
Decision Tree	splitter='best', criterion='entropy',min_sample_split=2, min_sample_leaf=1
Random Forest	n_estimators=100 , random_state=42
Extra Trees	default
Multi-layer Perceptron	default



Classification

Decision Tree

	precision	recall	f1-score
DoS	1.00	1.00	1.00
Normal	1.00	0.99	1.00
Probe	0.98	0.98	0.98
R2L	0.91	0.93	0.92
U2R	0.62	0.71	0.67
accuracy			0.99
macro avg	0.90	0.92	0.91
weighted avg	0.99	0.99	0.99

The Accuracy is:
0.9924589280904929

SVM

	precision	recall	f1-score
DoS	1.00	1.00	1.00
Normal	0.99	0.99	0.99
Probe	1.00	0.97	0.98
R2L	0.89	0.94	0.91
U2R	0.29	0.71	0.41
accuracy			0.99
macro avg	0.83	0.92	0.86
weighted avg	0.99	0.99	0.99

The Accuracy is:
0.9914085645030972

0.9964449232426609

	precision	recall	f1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	0.99	0.99	0.99
3	0.96	0.96	0.96
4	0.87	0.64	0.74
accuracy			1.00
macro avg	0.96	0.92	0.94
weighted avg	1.00	1.00	1.00

ExtraTrees

Random Forest

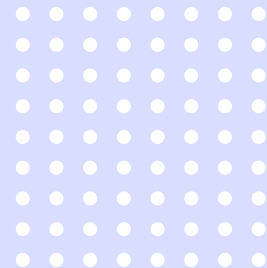
	precision	recall	f1-score
DoS	1.00	1.00	1.00
Normal	1.00	1.00	1.00
Probe	0.99	0.99	0.99
R2L	0.96	0.96	0.96
U2R	0.62	0.68	0.65
accuracy			1.00
macro avg	0.91	0.92	0.92
weighted avg	1.00	1.00	1.00

The Accuracy is:
0.9959601400484783

KNN

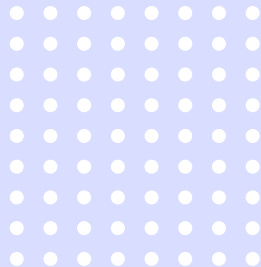
	precision	recall	f1-score
DoS	1.00	1.00	1.00
Normal	1.00	1.00	1.00
Probe	0.99	0.99	0.99
R2L	0.94	0.96	0.95
U2R	0.51	0.81	0.62
accuracy			1.00
macro avg	0.89	0.95	0.91
weighted avg	1.00	1.00	1.00

The Accuracy is:
0.9950983032588203



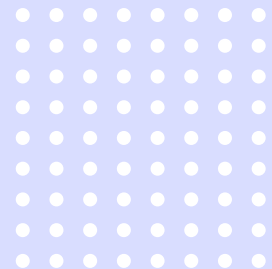
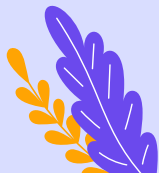
Classification

- We found that ET was best performing among all in terms of accuracy (99.64)
- KNN was performing really well on 'U2R' class as compared to others.
- SVM is weakest among all the used algorithm in terms of accuracy. Still, it performs well on U2R class.



Classification

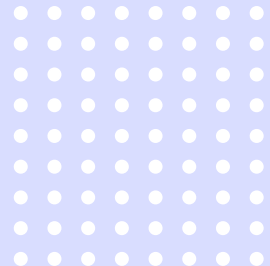
- So, we designed an ensemble model using:
 - ET
 - Knn
 - SVM
- 'Hard' voting was used.

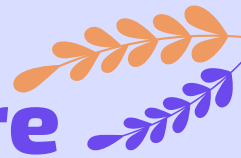


Results

- It was found that not only was our model performing great on whole dataset but also on 'U2R' class which was the least abundant in the dataset.

	precision	recall	f1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	0.99	0.99	0.99
3	0.95	0.96	0.95
4	0.93	0.60	0.72
accuracy			1.00
macro avg	0.97	0.91	0.93
weighted avg	1.00	1.00	1.00
The Accuracy is:			
0.9960678696471855			





Comparison with existing literature

Model	Accuracy	Minority class accuracy
Our Model	99.6	93
[2] Fitni et al.	98.8	84.3
[4] Karatas et al.	99.58	91.8
[6] Waskle et al.	96.78	No Multiclass given
[7] Mari et al.	91.6	78
[8] Liu et al.	99.1	No Multiclass given
[9] Xiaodong et al.	96.74	89



Comparison with smote and gan



precision recall f1-score support

DoS	1.00	1.00	1.00	13237
Normal	1.00	1.00	1.00	19346
Probe	0.99	0.99	0.99	3555
R2L	0.96	0.95	0.96	967
U2R	0.70	0.64	0.67	25

accuracy			1.00	37130
macro avg	0.93	0.92	0.92	37130
weighted avg	1.00	1.00	1.00	37130

The Accuracy is:
0.9959601400484783

With Smote (old)

The Accuracy is:
0.9321303528144358

	precision	recall	f1-score
DoS	0.96	0.93	0.94
Normal	0.96	0.96	0.96
Probe	0.91	0.87	0.89
R2L	0.55	0.71	0.62
U2R	0.14	0.96	0.25
accuracy			0.93
macro avg	0.70	0.88	0.73
weighted avg	0.94	0.93	0.94

With GAN

0.9935900888769189

	precision	recall	f1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	0.99	0.99	0.99
3	0.89	0.94	0.92
4	0.71	0.40	0.52
accuracy			0.99
macro avg	0.92	0.87	0.88
weighted avg	0.99	0.99	0.99

Without Smote (old)

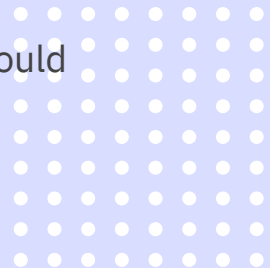
	precision	recall	f1-score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	0.99	0.99	0.99
3	0.95	0.96	0.95
4	0.93	0.60	0.72
accuracy			1.00
macro avg	0.97	0.91	0.93
weighted avg	1.00	1.00	1.00

The Accuracy is:
0.9960678696471855

With SMGAN(new)

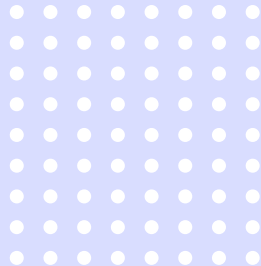
Conclusion & Future Work

- It was found that SMGAN performed better than simple SMOTE and GAN applied individually.
- It was found that our model had better evaluation metrics when compared to other works.
- It was also better in classifying minority class ('U2R') as compared to others.
- The main for the success of this model can be attributed to our devised approach for generating new samples (SMGAN), Hybrid feature selection and our ensemble model used for classification.
- For future works, Deep Learning and Reinforcement Learning based approaches should be used while classification.



New Work

- Optimized the ensemble model by including Extra Trees Algorithm instead of Random Forest.
- Concluded the research and planning to submit it in a conference.

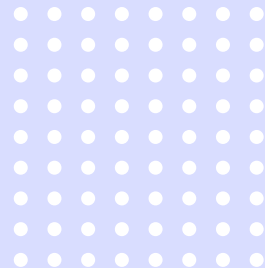


References

All the code is implemented in python using sklearn and imblearn library.

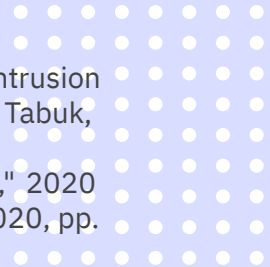
Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp.

```
@article{JMLR:v18:16-365,  
author = {Guillaume Lemaire and Fernando Nogueira and Christos K. Aridas},  
title   = {Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine  
Learning},  
journal = {Journal of Machine Learning Research},  
year    = {2017},  
volume  = {18},  
number  = {17},  
pages   = {1-5},  
url     = {http://jmlr.org/papers/v18/16-365.html}  
}
```



References

- [1])Alshamy, Reem & Ghurab, Mossa & Othman, Suad & Alshami, Faisal. (2021). Intrusion Detection Model for Imbalanced Dataset Using SMOTE and Random Forest Algorithm. 10.1007/978-981-16-8059-5_22.
- [2])Fitni, Q. R. S., & Ramli, K. (2020). Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems. 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT).
- [3])Naveed, Muhammad & Arif, Fahim & Usman, Syed & Anwar, Aamir & Hadjouni, Myriam & Elmannai, Hela & Hussain, Saddam & Ullah, Syed Sajid & Umar, Fazlullah. (2022). A Deep Learning-Based Framework for Feature Extraction and Classification of Intrusion Detection in Networks. Wireless Communications and Mobile Computing. 2022. 1-11. 10.1155/2022/2215852.
- [4])G. Karatas, O. Demir and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," in IEEE Access, vol. 8, pp. 32150-32162, 2020, doi: 10.1109/ACCESS.2020.2973219.
- [5])A. Abdullah ALFRHAN, R. Hamad ALHUSAIN and R. Ulah Khan, "SMOTE: Class Imbalance Problem In Intrusion Detection System," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-5, doi: 10.1109/ICCIT-144147971.2020.9213728.
- [6])S. Waskle, L. Parashar and U. Singh, "Intrusion Detection System Using PCA with Random Forest Approach," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 803-808, doi: 10.1109/ICESC48915.2020.9155656.



References

- [7] Mari, A.-G.; Zinca, D.; Dobrota, V. Development of a Machine-Learning Intrusion Detection System and Testing of Its Performance Using a Generative Adversarial Network. *Sensors* **2023**, 23, 1315. <https://doi.org/10.3390/s23031315>
- [8] Xiaodong Liu, Tong Li, Runzi Zhang, Di Wu, Yongheng Liu, Zhen Yang, "A GAN and Feature Selection-Based Oversampling Technique for Intrusion Detection", *Security and Communication Networks*, vol. 2021, Article ID 9947059, 15 pages, 2021. <https://doi.org/10.1155/2021/9947059>

