# Annon - Network Intrusion Detection Challenge

Sridhar Ramasamy
Grad. Student Researcher
Colorado State University,
Fort Collins, CO


# Introduction

- Machine learning is the science of getting computers to act without being explicitly programmed
- With an estimate of around 2.3 trillion GB of data being created each day, the question before us how well do we know the data ?
- What kind of insights can be drawn ?
- Can the data be visualized ?
- Further can we predict the answers for similar pattern that we might encounter in the future ?
- Yes, there are answers for all this
- This response to this challenge will comprise of answers to the above questions.

# Overview

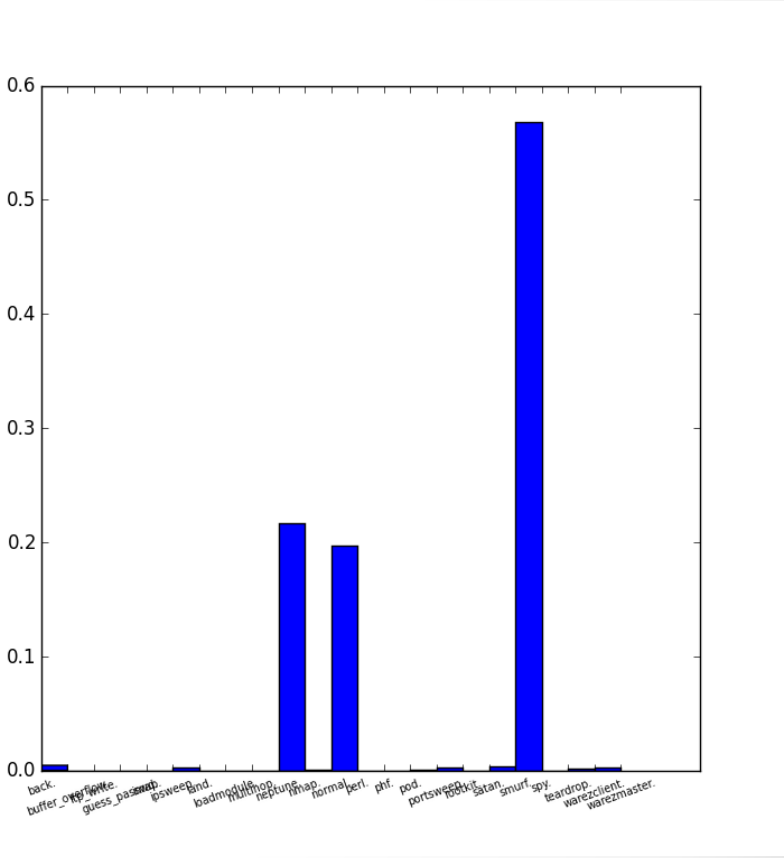
- The presentation is divided into following sections
  - Overview of Dataset
    - Histogram of attack types
    - The Rank of a dataset/Singular Value Decomposition
    - Correlation of the features
  - The challenge
    - Multi-class classification
    - Support Vector Machine
  - Results
    - Training SVM – choosing the parameters
    - Accuracy of the approach - Accuracy and F1 Score
    - Possible approaches

# Dataset

- The dataset is historical data captured in the network.
- The captured data shows both normal connections and intrusion attack.
- The data comprises of two sets. The training dataset and the test dataset.
- The dataset consists of 41 features and the last field is the "*attack type*".
- **What are the insights that can be obtained from this dataset ?**

# Histogram

- The histogram of the training dataset is shown.
- It can be seen that over 55% of attacks are smurf attacks followed by neptune attack with approximately 22%.
- Around 20% of the connections are found to be normal.

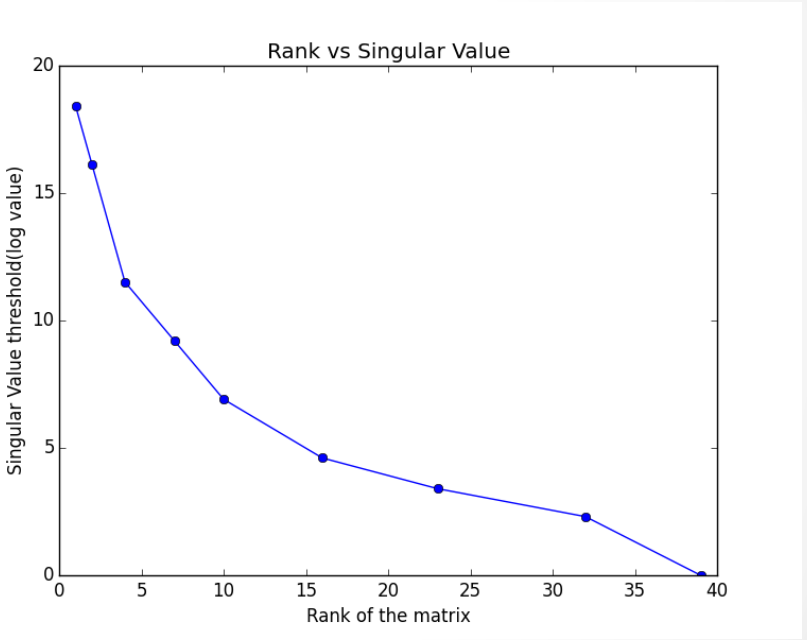


# Rank of matrix

- The training data can be converted to a matrix of (MxN) dimension, where "M" is the number of samples and "N" is no of features.
- What is a rank of a matrix ? And what is low rankness?
  - Rank of a matrix is the "**number of linear independent rows**"
  - Suppose, if the rank of a (1000x1000) matrix "A" is 5 and the rank of a (10x10) matrix "B" is also 5 then we can say that **"A" is low in rank compared to "B"**
- Why should we bother about this ?
  - Rank of a matrix(non-zero singular values) gives the significance of the number of principal components and it will aid in **dimensionality reduction**[1].
  - If we know the few linearly independent rows of a low rank matrix then we have a effective method to reconstruct the entire matrix **– low rank approximation**.[1].
  - Robust PCA and extended Robust PCA is about matrix decomposition and completion[1][2].

# Singular Value

- To obtain the singular values, one common approach is Singular Value Decomposition[4]. The time complexity is $O(\min(mn^2, nm^2))$ (depending on the no of samples and features)
- By setting a singular value threshold, numpy matrix rank method gives another way to infer the rank



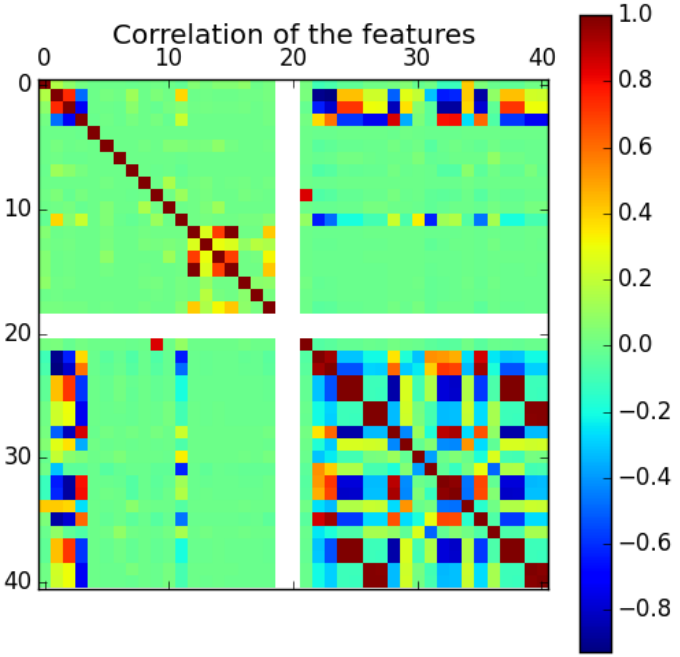
# Singular value contd.

- These singular value leads to some good conclusions
- We can see that if we set a threshold of 20 we will be able to reduce the dimension by around 10.
- These features would be redundant, meaning that they will convey the same information.

| Singular Value threshold | Rank |
|---|---|
| 100000000 | 1 |
| 10000000 | 2 |
| 100000 | 4 |
| 10000 | 7 |
| 1000 | 10 |
| 100 | 16 |
| 30 | 23 |
| 20 | 28 |
| 10 | 32 |
| 1 | 39 |

# Correlation

- The correlation of 41 features is calculated by Pearson coefficient.
- Values close to +1 or -1 (extremes) denotes the features are linearly dependent.
- Values closer to ‘0’ reflects that features are not linearly dependent.
- For a couple of features, the standard deviation is 0, hence correlation coefficient is NaN



# Correlation contd.

- Plot shows that there are few features that appear to be well correlated.

- This is also conveying the same answer as that of rank vs singular value plot and the table.
- While reducing the dimensions, **all** the significant singular values should be taken into account.
- In that sense the "**rank can be considered to be ~30**"

# Intrusion detection challenge

- The model to be created should be trained on the given training set and evaluated on the test set.
- The training dataset contains 22 different types of attack detected and the normal one.
- The field, protocol_type , service, flag, attack_type contains strings that are encoded with Label Encoder of scikit-learn.
- The dataset is normalized between values of 0 and 1.
- The model should be able to classify the attacks apart from detecting them.

# Contd.

- The test dataset on the other hand contains few extra type of attacks, so after prediction these extra type of attacks get misclassified because of the fact that the model has not been trained with it.

# Multi-class Classification & SVM

- The model should be able to classify the type of attack apart from detecting the attack and hence multi-class classification is used for this.
- Support Vector machine SVC with RBF kernel has been used for this.
- The accuracy of the approach depends upon the two parameters ‘C’ and ‘gamma’.
- ‘C’ is the penalty parameter for misclassification of a sample and ‘gamma’ denotes the influence of single training example.

# SVM parameters

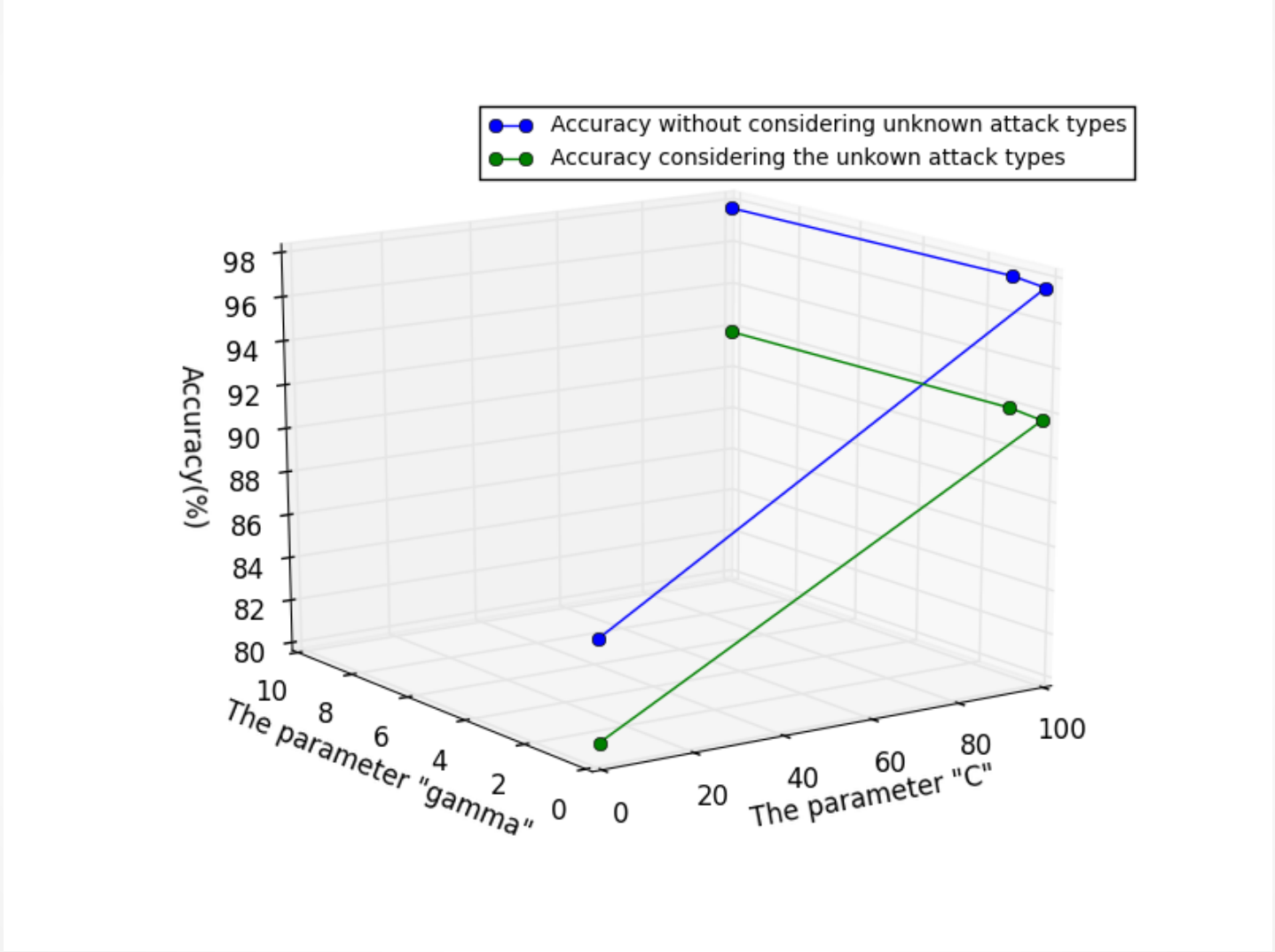
- The C and gamma parameters have been changed and results have been obtained.
- Label encoder was used to encode fields that were strings.
- The model was trained only with the attack types of 'training dataset'
- The **accuracy** of correct prediction for the entire vector.
- The **F1 Score** has also been obtained for classifying (binary) it as any of **attack types** or **normal.**
- Results have been obtained for
  - C=1, gamma=0.0243(auto)
  - C=100,gamma=0.0243
  - C=100,gamma=1
  - C=100,gamma=10

# Results

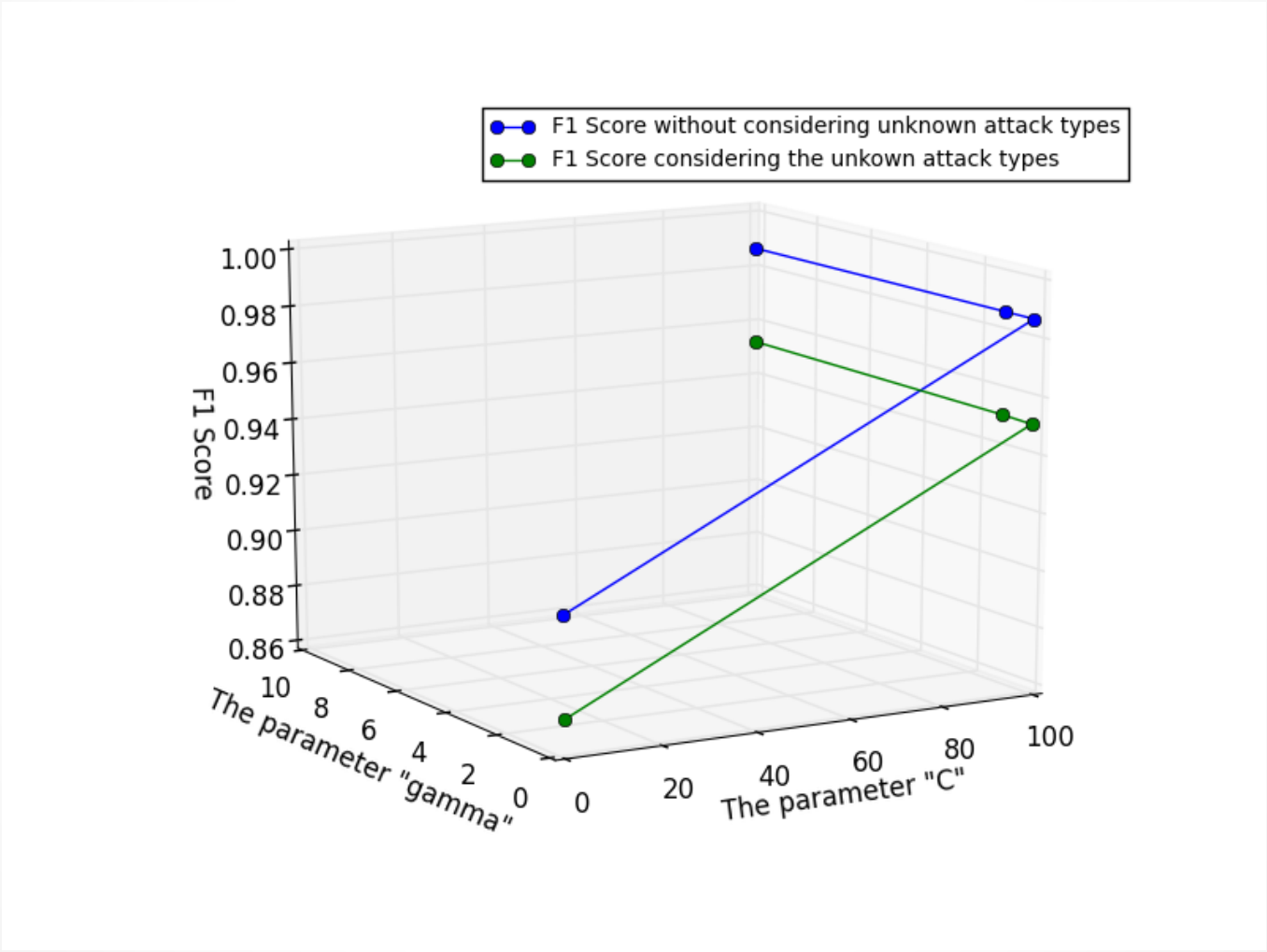
- Accuracy is calculated as follows,
  - $Accuracy = \frac{Number\ of\ correctly\ predicted\ entries}{total\ number\ of\ entries}$

- F1 Score is calculated as follows,
  - $F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$
  - $Precision = \frac{TP}{TP+FP}$
  - $Recall = \frac{TP}{TP+FN}$
  - Where TP, FP, FN are True positive, False positive and False Negative respectively

# Results - Accuracy



# Results – F1 Score



# Results

| Parameters | Accuracy without unknown attack types |
|---|---|
| C=1,g=auto | 85.822% |
| C=100,g=auto | 97.550% |
| C=100,g=1 | 97.72% |
| C=100,g=10 | 97.67% |

| Parameters | F1 Score without unknown attack types |
|---|---|
| C=1,g=auto | 0.90431 |
| C=100,g=auto | 0.98662 |
| C=100,g=1 | 0.98667 |
| C=100,g=10 | 0.9869 |

| Parameters | Accuracy with unknown attack types |
|---|---|
| C=1,g=auto | 80.65% |
| C=100,g=auto | 91.676% |
| C=100,g=1 | 91.841% |
| C=100,g=10 | 91.79% |

| Parameters | F1 Score with unknown attack types |
|---|---|
| C=1,g=auto | 0.8691 |
| C=100,g=auto | 0.9507 |
| C=100,g=1 | 0.95113 |
| C=100,g=10 | 0.9525 |

# Further work

- The train dataset and test dataset can be combined together and it can be **cross validated** by splitting it into different chunks for the same SVM.
- Neural Networks can also be used for this.
- The redundant features can be avoided with help of dimensionality reduction(SVD).

# Reference


1. Space-time signal processing for distributed pattern detection in sensor networks - Randy Paffenroth et al.
2. Robust Principal Component Analysis? Candes et al.
3. http://www.ibmbigdatahub.com/infographic/four-vs-big-data
4. Topology Maps and Distance-Free Localization from Partial Virtual Coordinates for IoT Networks – Anura P Jayasumana, Randy Paffenroth, Sridhar Ramasamy