Executive Summary – Network Intrusion Detection

Introduction

For the purpose of our task, we were provided a sample of the KDD Cup 1999 dataset meant for use to create a neural network model designed for detecting intrusions or attacks on a computer network. The neural network model should be capable of distinguishing between 'bad connections' (intrusions or attacks), and 'good connections' (normal connections). **Our work for this task includes:**

- The pre-processing and analysis of the dataset including normalization and dimensionality reduction, more details on the next chapter.
- The development and testing of the **PCA (Principal Component Analysis)** and **LDA (Linear Discriminant Analysis)** along the **GaussianNB (Naïve Bayes)** and **(SVM) Support Vector Machine Classifiers**.
- The testing of the PCA, LDA techniques with different configurations of the SVM Classifier to determine best prediction model.

Other work includes: creating a CNN (Convolutional Neural Network) for testing against the methods implemented, at the moment the CNN model is not fully functional and requires further work.

Data Pre-processing & System Development

Initial analysis on the dataset found a total of **494021 data points** with **42 features**, after dropping the duplicates found (**348435**), the new shape is **(145586, 42).** Two important observations made during the data analysis were: the dataset being *highly imbalanced* with some classes not being well represented and several highly *correlated features*. Two techniques were used to deal with the dimensionality of the data, PCA and LDA, during testing we found PCA to need **30 components** to represent the entire data. When choosing our train/test ratio, throughout our research we found the ideal combination to be **training – 70%** and **testing – 30%. The final dimensions for the test/train data were:**

Training Set:	**Testing Set:**
(101910, 41)	**(43676, 41)**

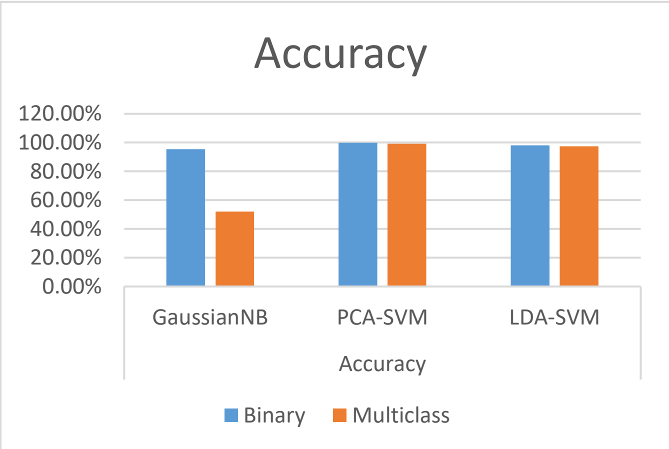
The data was then normalized using the **Standard Scaler** in python, and two classification approaches were designed for our models: ***Binary*** and ***Multiclass***. The dataset has the following categories: ***back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster***. We have assigned those categories for two classes in Binary classification, and five classes in Multiclass.

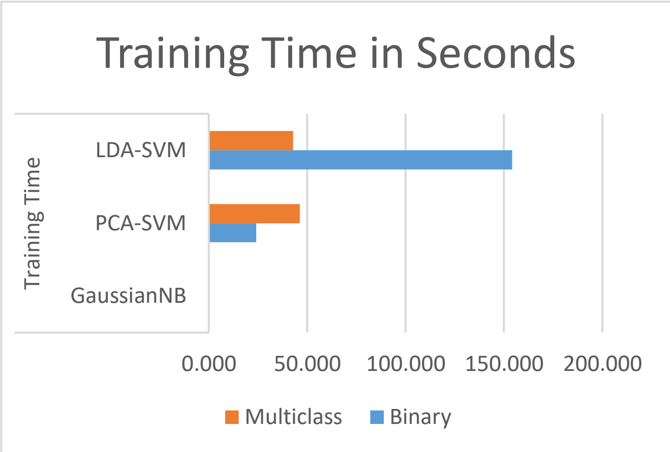
- **Binary** – *Attack (57754) & Normal (87832)*
- **Multiclass** – *DoS (54572), Probe (2131), R2l (999), U2r (52), Normal (87832)*

During development we have also tested the **SVM Classifier** with both **PCA** and **LDA** using the *Linear* and *RBF* kernels to determine best prediction model. RBF has provided best accuracy results and relatively low training times.

Results

Highest results were provided by the PCA-SVM model with the RBF kernel, with accuracy of **99.78%** for the binary classification and **99.15%** for the multiclass. Generally, all models, except GaussianNB, performed well with accuracy above 95% in both binary and multiclass. The GaussianNB registered a relatively good score for binary but was outperformed significantly for multiclass.





Accuracy: **GaussianNB** – *Binary **(95.33%)**, Multiclass **(52.08%)**;* **PCA-SVM** – *Binary **(99.78%)**, Multiclass **(99.15%)**;* **LDA-SVM** – *Binary **(98.01%)**, Multiclass **(97.48%)**.*

Training Time: **GaussianNB** – *Binary **(0.109s)**, Multiclass **(0.125s)**,* **PCA-SVM** – *Binary **(24.114s)**, Multiclass **(46.252s)**;* **LDA-SVM** – *Binary **(154.108s)**, Multiclass **(42.856s)**.*

Conclusion

No metric can replace real-world conditions. While we have managed to obtain surprising results, for better prediction and further testing more data is needed. The PCA-SVM is the best model for intrusion detection with only **96** mislabelled points out of **43676**. Future work can include the implementation of the CNN and tackling more classes for classification.