IMAT5235 – Artificial Neural Networks (ANN)

Network Intrusion System

A Comparative Study of Machine Learning Techniques for Intrusion Detection

Robert Barbulescu
MSc Intelligent Systems & Robotics
De Montfort University

Problem Statement

- A neural network model is to be designed for detecting intrusions or attacks on a computer network.
- As a minimum the neural network model should be capable of distinguishing between 'bad connections' (intrusions or attacks), and 'good connections' (normal connections).
- A database ***(KDD Cup 1999)*** that contains several intrusions simulated in a military network environment has been provided for the purpose of this task.
- **Note:** for the purpose of this work, we will work with a simplified version of the database, a dataset sample of only 10 % of the entire dataset is used.

Our Approach

- The initial stage when developing the neural network model was performing data analysis on our dataset.
  - Important factors that we are looking for are:
    - *How is the data distributed ?*
    - *Overall structure of the data ?*
    - *Any correlated or redundant information ?*
    - *Any duplicate data ?*
- After data processing occurred, a decision was made to implement and test two classifiers in combination with two dimensionality reduction techniques.
  - Those models are to perform intrusion detection using two main approaches:
    - **Binary classification**
    - **Multiclass classification**
- The following models have been implemented:

I. **Gaussian Naïve Bayes with Principal Component Analysis (PCA)**
II. **Support Vector Machine with Principal Component Analysis (PCA)**
III. **Support Vector Machine with Linear Discriminant Analysis (LDA)**

KDD Cup 1999 Dataset

- The KDD is a specially designed database containing a TCP dump of around five million connection record, for our purposes we are using only 10% of that.
  - ***'connection'*** - a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol.
  - each connection is tagged as either normal, or as an attack, with exactly one specific attack type [1].
- Attacks belong to the following main categories:
  - **Denial of Service (DoS)**: an attack in which an adversary directed traffic requests to a system in order to make the computing or memory resource too busy or too full to handle legitimate requests and in the process, denies legitimate users access to a machine.
  - **Probing Attack (Probe)**: probing network of computers to gather information to be used to compromise its security controls.
  - **User to Root Attack (U2R)**: a class of exploit in which the adversary starts out with access to a normal user account on the system (gained either by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.
  - **Remote to Local Attack (R2L)**: occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

[1] 'KDD-CUP-99 Task Description'. http://kdd.ics.uci.edu/databases/kddcup99/task.html (accessed May 13, 2021).

Data Preprocessing & Analysis

- When extracting and reading our dataset, we obtained the following information:

Data Points:	494021
Features:	42
Initial Dataset Shape:	(494021, 42)

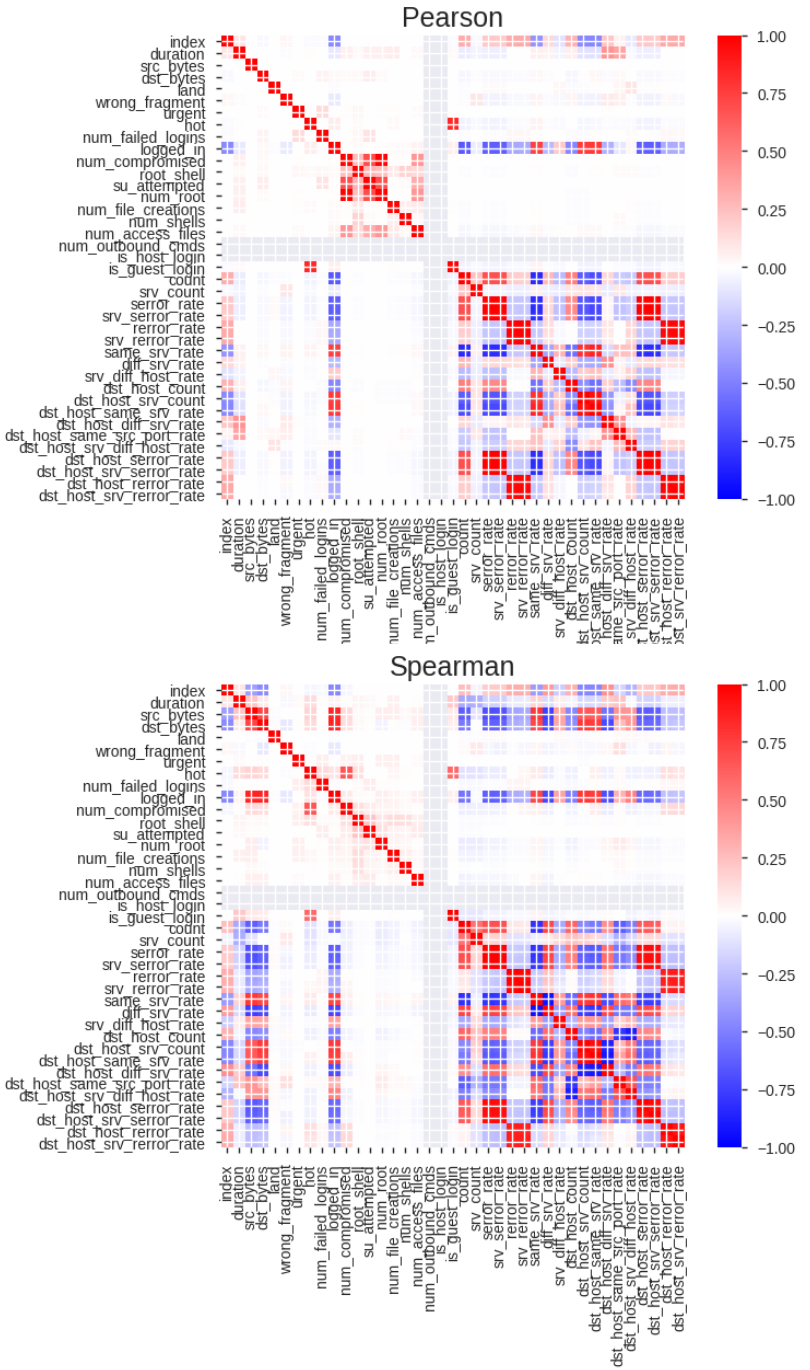
- Our next steps was looking for any NULL values and eliminating any duplicates:

Existing NULL values:	0

- Duplicates found: **348435**

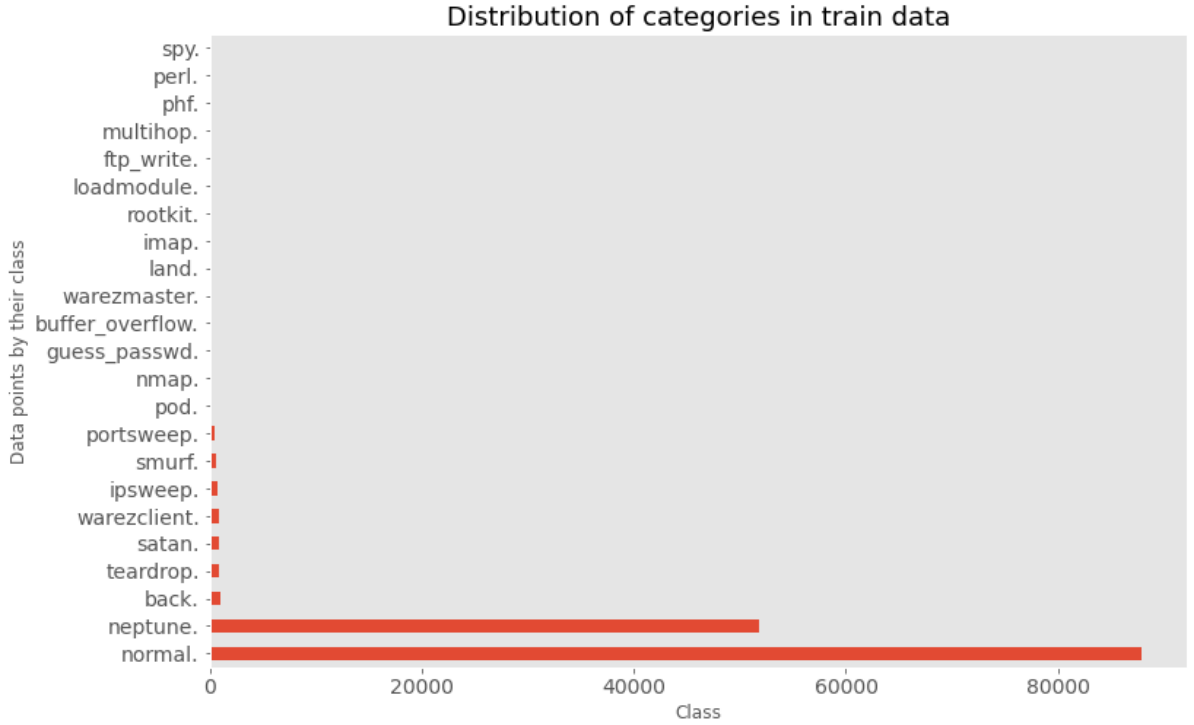
New data shape is:	(145586, 42)

- A full and comprehensive analysis of the dataset was performed in python using the *pandas* module and provided us with the the two correlations plots on the righten side.
  - *Looking at the diagram we can see that several features are highly correlated.*



Feature Correlations & Distribution of Categories

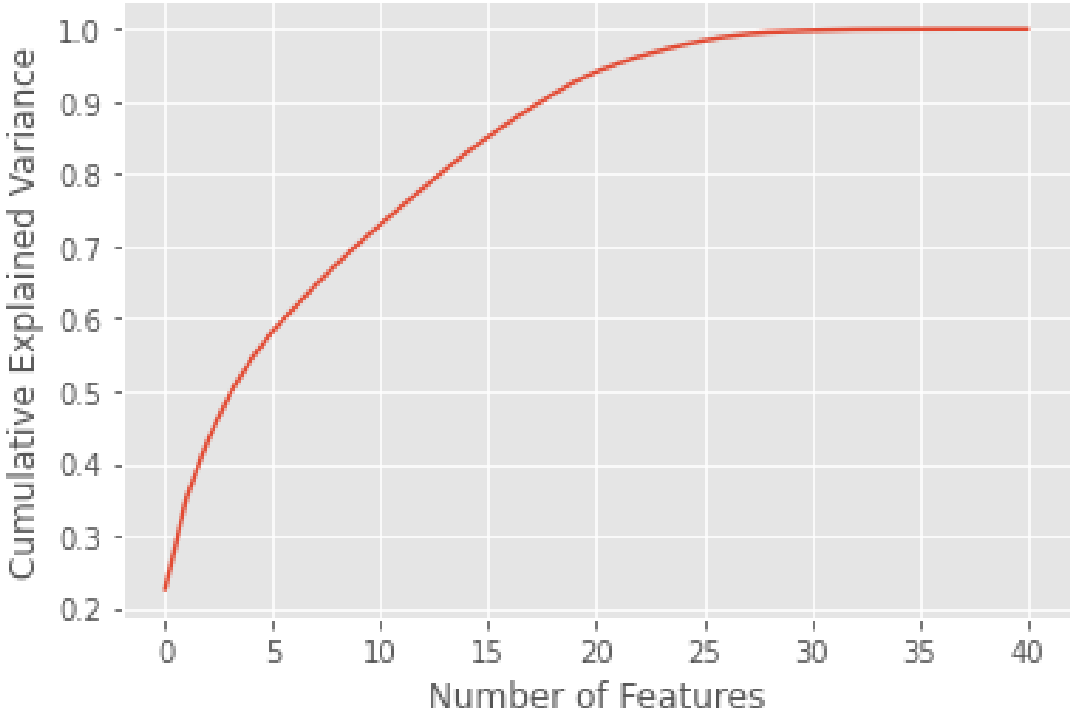
- ***dst_host_serror_rate***
  - Highly correlated
  - This variable is highly correlated with ***srv_serror_rate*** and should be ignored for analysis
  - Correlation = *0.99512*
- ***dst_host_srv_serror_rate***
  - Highly correlated
  - This variable is highly correlated with ***dst_host_serror_rate*** and should be ignored for analysis
  - Correlation = *0.9959*
- ***dst_host_rerror_rate***
  - Highly correlated
  - This variable is highly correlated with ***srv_rerror_rate*** and should be ignored for analysis
  - Correlation = *0.96737*
- ***dst_host_srv_rerror_rate***
  - Highly correlated
  - This variable is highly correlated with ***dst_host_rerror_rate*** and should be ignored for analysis
  - Correlation = *0.9715*



- The highest data point percentage comes from the class ***normal*** which is the good connection or non-intrusion category.
- The highest number of data points for the intrusion or bad connection category are provided by ***neptune*** and ***back.***
- The remaining have the smallest number of data points with some reaching less than 10 per class.
- The dataset is highly imbalanced with some classes being not well represented, for a better prediction more data is needed.

Overview of Models & Techniques

- **Dimensionality Reduction**
  - **PCA** - *dimensionality reduction method*, discovers a set of projection vectors which retains most properties of the original data.
  - **LDA** - *traditional approach in pattern recognition,* technique looks for a linear combination of features or linear transformation that allows for the separation of two classes or objects.
- **Classifiers**
  - **SVM** - *learning machine method,* implements the basic idea of non-linear transform so that the sample space will be linearly separable after changing its characteristics.
    - SVM performs separation or discrimination between two classes.
  - **Naïve Bayes** - *supervised learning algorithm*, a probabilistic classifier that predicts on the basis of the probability of an object.
    - ***Advantage:*** simple and effective, allows for fast predictions.



- When setting the parameters for our PCA technique, we looked at the cumulative variance which shows that around 30 components are needed for 100% variance.
- That is the number of features that can represent the entire dataset.
- Our implementations use 30 components across all techniques and classifiers.

System Development

a) When implementing our models, we have used PCA with 30 components as a data pre-processing step.

b) The train and test data have been structured in a 70/30% ratio, this was found to be the ideal proportion during our research.

Training Data: **101910**	Testing Data: **43676**

a) Feature scaling has been applied to normalize our data.

b) Two scaling methods were initially considered, ***Min Max Scaler*** and ***Standard Scaler,*** but as the Min Max does not perform well with Gaussian distributions a decision has been made to use the Standard Scaler.

c) Final dimensions are presented below:

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

d) Our data has the following features: ***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.***

e) Two approaches to classifying our data:

- **Binary** – *two classes (Normal & Attack)*
- **Multiclass** – *five classes (DoS, Probe, R2l, U2r, Normal)*

Binary Dataset Distribution

Attack:	**57754**
Normal:	**87832**

Multiclass Dataset Distribution

DoS:	**54572**
Probe:	**2131**
R2l:	**999**
U2r:	**52**
Normal:	**87832**

- Looking at the table above we can see that *DoS* and *Normal* have the highest numbers, with the remaining classes being considerable smaller.
- This can potentially cause issues during implementation as the dataset is not well distributed.
- One potential factor can be a decrease in accuracy for such classes.

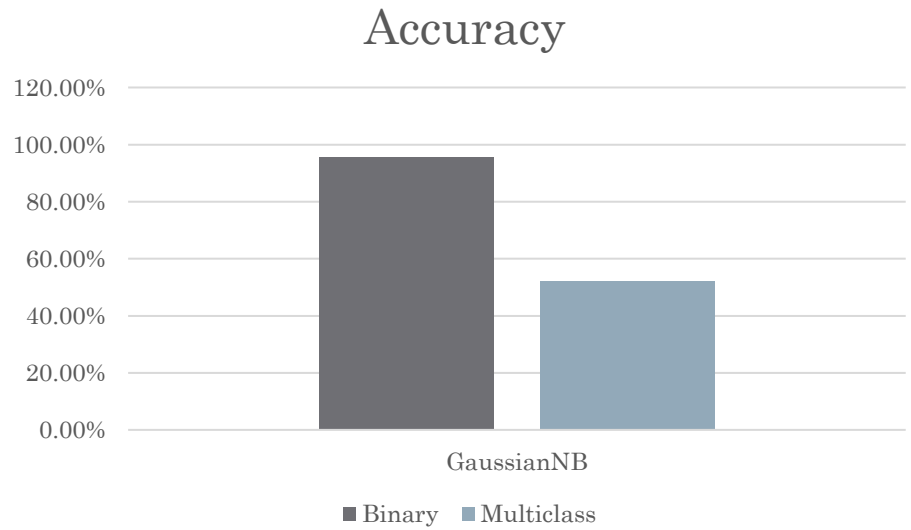
Model Testing – GaussianNB

Accuracy:

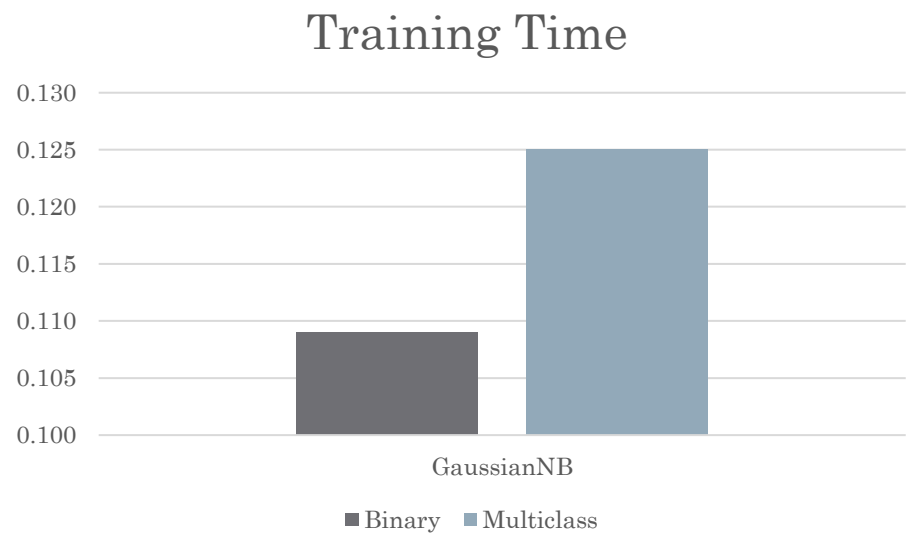
	Binary	Multiclass
GaussianNB	95.33%	52.08%

Training Time:

	Binary	Multiclass
GaussianNB	0.109	0.125



- The model appears to perform well in the binary classification but severely underperforms for the multiclass classification.
- Similarly, the training time is lower when performing binary classification than the multiclass.
- ***The best results registered for <u>GaussianNb</u> are 95.33% accuracy with 0.109 seconds.***



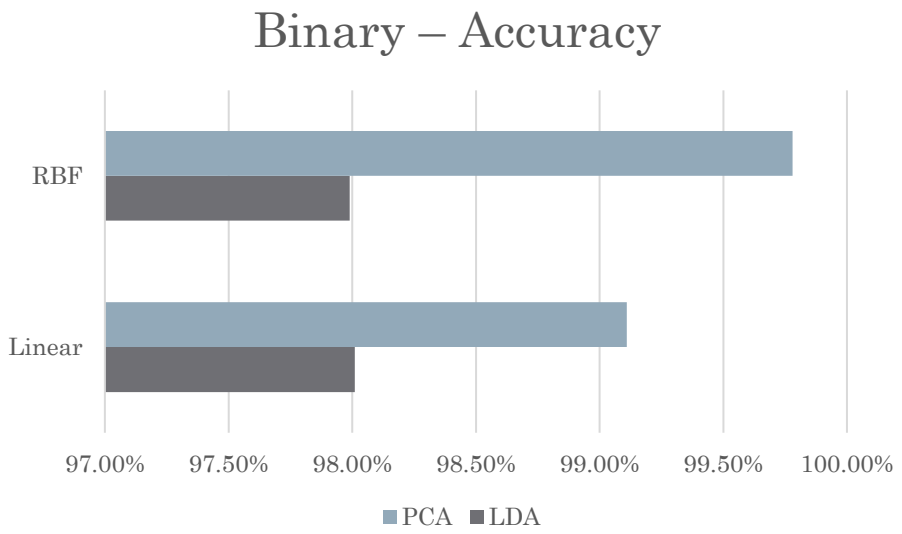
Model Testing – SVM

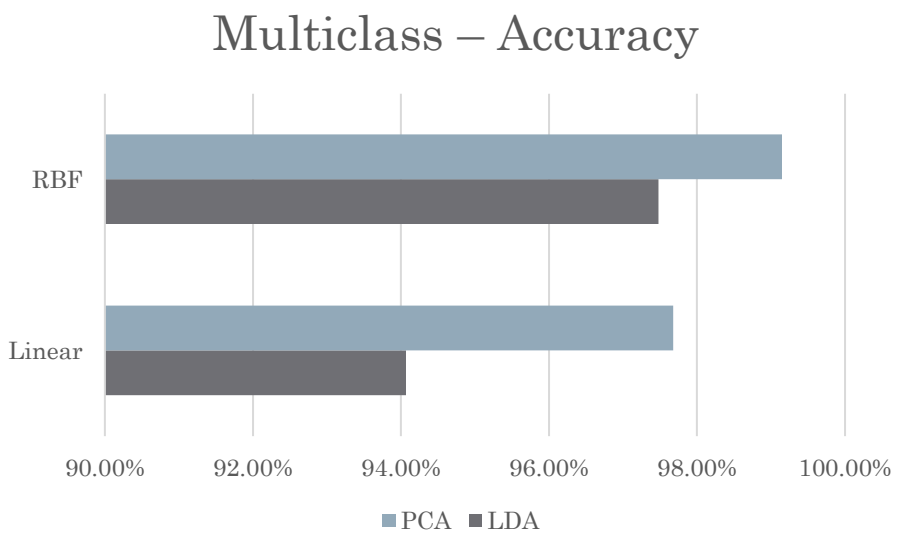
- Unlike the Gaussian Naïve Bayes model, The SVM model has been tested using two dimensionality reduction, as well as using two different kernel functions to determine the best parameters for our system.
- The results from all our testing are provided below:

Binary		
	Linear	RBF
LDA	98.01%	97.99%
PCA	99.11%	99.78%

Multiclass		
	Linear	RBF
LDA	94.07%	97.48%
PCA	97.68%	99.15%

- Looking at the accuracy results above we can notice that the highest accuracy, achieved by the binary approach, is the PCA-SVM with a RBF kernel at 99.78%, the multiclass also appears to perform significantly well, a considerable improvement from GaussianNB with most techniques reaching above 90%.
- Generally, the LDA technique appears to be outperformed by the PCA regardless of the kernel.





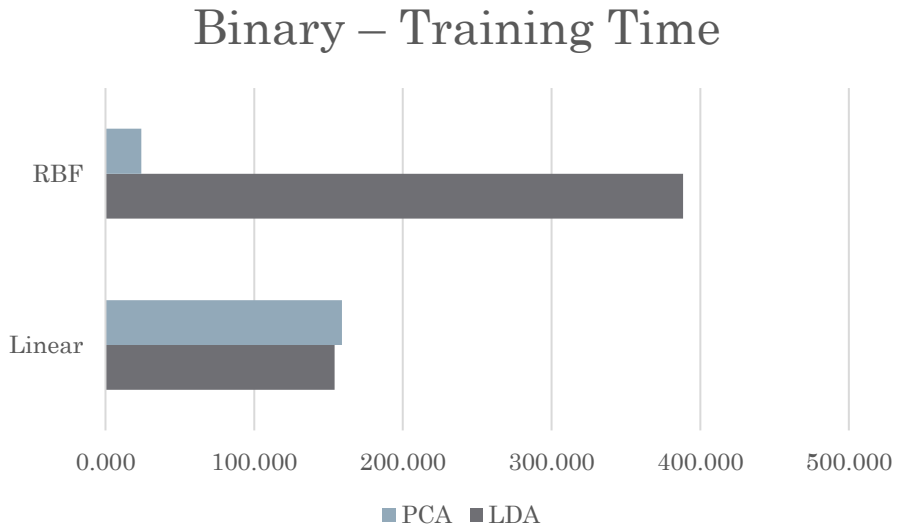
Model Testing – SVM

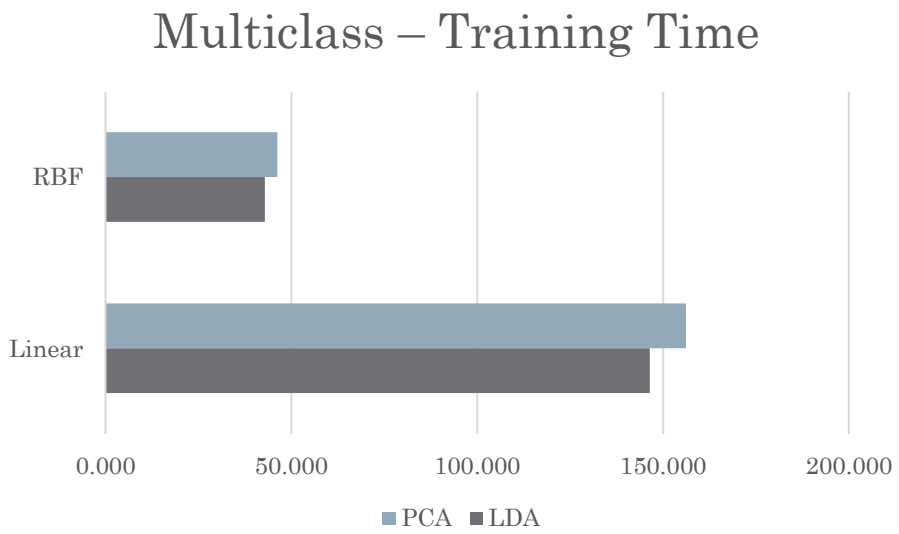
- In terms of training time, we can see from the graphs and tables below that the binary PCA with the RBF kernel has the lowest training time at just 24 seconds.

Binary		
	Linear	RBF
LDA	154.108	388.447
PCA	159.067	24.114

Multiclass		
	Linear	RBF
LDA	146.455	42.856
PCA	156.178	46.252

- In most cases the RBF kernel provides lower training time, except the binary LDA, which registers the highest training time among all techniques.
- ***The best results registered for <u>SVM</u> are 99.79% accuracy with 24.114 seconds.***





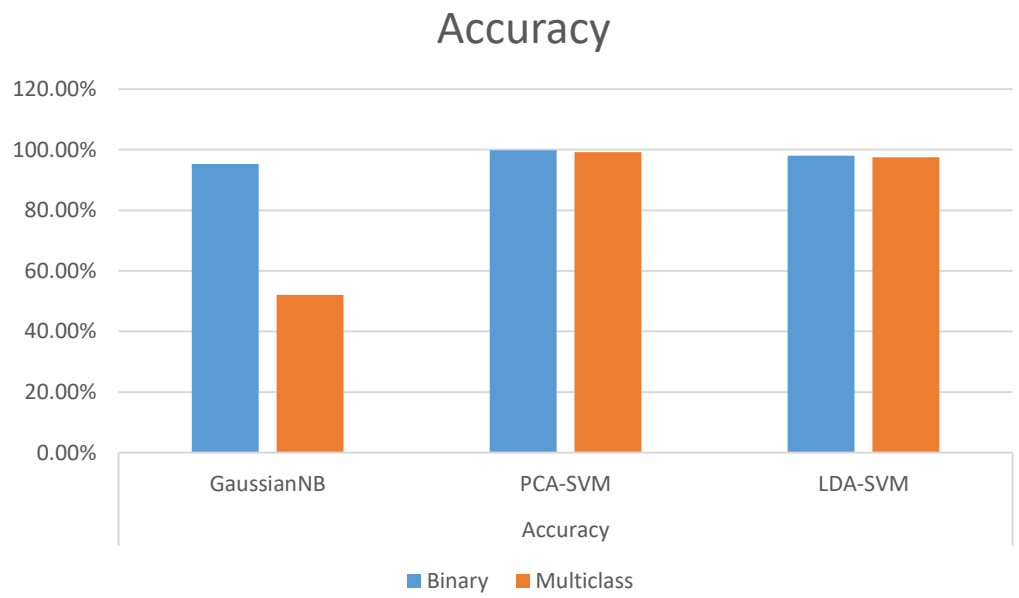
Contrast of Results

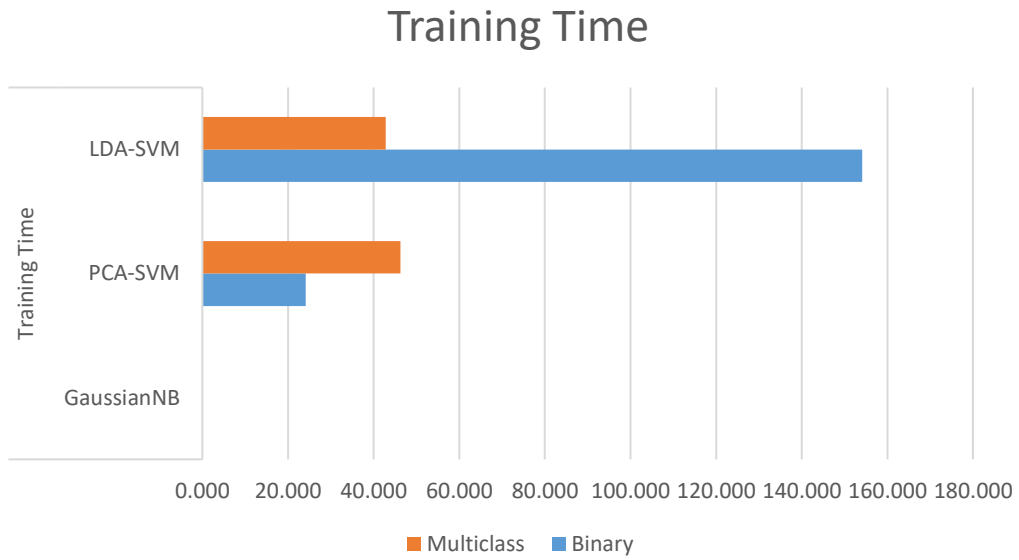
- Looking at all systems and techniques implemented, best results in both binary and multiclass are provided by the PCA-SVM system with the RBF kernel, registering up to 99.78% in binary classification and 99.15% in multiclass, as well as relatively good training times with a maximum of 46 seconds for multiclass.
- All models except GaussianNb appear to perform well in both multiclass and binary with accuracy being generally over 95%.
- Lowest training time is by far the GaussianNB, with 0.125 seconds being the maximum in the multiclass classification. Unfortunately, it is also the lowset accuracy across al models.

Method	Accuracy		
	GaussianNB	*PCA-SVM*	*LDA-SVM*
Binary	95.33%	99.78%	98.01%
Multiclass	52.08%	99.15%	97.48%

Method	Training Time		
	GaussianNB	*PCA-SVM*	*LDA-SVM*
Binary	0.109	24.114	154.108
Multiclass	0.125	46.252	42.856

- Our best model, the PCA-SVM (RNF Kernel) has the number of mislabelled points out of a total **43676** points is **96.**





Conclusion

- No metric can replace real-world conditions.
- When using the KDD99 dataset with the parameters presented in the previously we can achieve a score of above 99% accuracy.
- The PCA-SVM implementation has performed best with a score of 99.78%, followed by the LDA-SVM with 98.01% and GaussianNB with 95.33%.
- Some important considerations are the classes in our dataset, while we have achieved high detection scores the classes are not well represented, meaning for more complex systems with more classes the prediction will have significantly lower results. More data is needed.
- Future work can include testing our models against a more complex system based on a CNN or creating a framework using the models presented for better detection.