INTRUSION DETECTION CLASSIFICATION MODEL

JULIAN PEDRAZA - 500929362

CKME 136 – CAPSTONE

1. ABSTRACT

1.1. Dataset History

The dataset is a sample of about seven weeks of network traffic, the product of a simulation of typical U.S. Military entity LAN. The intention of this was to research deeper into fraud and intrusion detection. The simulation includes 22 types of attacks, delimited in 4 types of categories.

The dataset was first presented in the KDD Cup 1999 contest, it contains 4.5 million records 40 attributes and the class attribute.

https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data

1.2. Problem

Build a classification model that can detect the bad connections (also called attacks) and the good connections

1.3. Question for Analysis

- Is it possible to determine the type of intrusion, based on the characteristics of the connection to the network?
- Is it possible to define which are the more relevant attributes that determine if a connection is bad or good?
- What is the classification algorithm that performs better?

2. LITERATURE REVIEW

2.1. Reference datasets

Intrusion Detection Systems (IDS) have been an important part of the development of computer networks and the internet itself. Since the last century, various datasets have been released and have been the product of study and the topic in several papers, mainly related to classification modeling.

- **CIDDS-001:** Coburg Network Intrusion Detection Dataset was originally presented in 2017, it is a flow-based benchmark data set for intrusion detection. This dataset emulates a small business environment, various clients and typical servers, in which users are surfing the web, sending emails, prepare documents and print. This new dataset was created as an alternative for obsolete datasets for IDS.
- **DARPA 1999:** This dataset can be considered as the baseline for any research, but any effort to make it more real will help researchers to keep working towards the evolution of the Intrusion Detection Systems.

Other researchers were not satisfied with the procedure used when the dataset was built, mainly created using simulated resources and attacks implemented via scripts, additionally, the dataset was enhanced by injecting new data from a single host.

- **KDD 99:** This dataset was presented at the KDD Cup an annual competition in data mining. This is one of the most widely used datasets for the evaluation of anomaly detection. Regardless of what the detractors and supporters think about the dataset, it is still a reference and starting point for analysis and evaluation.
- **NSL-KDD:** The NSL-KDD dataset, was created as an alternative to the KDD 99 dataset, mainly designed to solve a big proportion of the issues discovered, this paper additionally shows a historical comparison against DARPA' 98, and their common problems, statistical observations on the KDD 99 dataset and some solutions that will help researchers better detected the attacks.

As part of the literature review for this project, various papers were revised, and their methodologies were used as the basis for this work. A link will be available in the Github Repository.

2.2. Conceptual approach

An Intrusion Attack is a security incident that affects an information system, that can be provoked by members of an organization, or outsiders with access to the network, via use of social engineering attacks, while taking advantage of an incautious member of an organization that will provide access to data and information systems without even noticing.

The following are the types of attacks included as part of the KDD 1999 dataset:

- **Denial of Service (DoS):** The attacker sends useless load (large volume of packets) to the target or victim, overloading its system and consuming all the available resources, with the intention of make the system to fail until it turns unavailable for legitimate data and users.
- **User to Root Attack (U2R):** The attacker has access to the victim machine or user account on the system and from there is an opportunity to look for vulnerabilities and to gain superuser privileges[1].
- **Remote to Local Attack (R2L):** The attacker or hacker can send packets over the network to the victim machine and tries to get access to the device via potential vulnerabilities in the system.
- **Probe or Probing:** The attacker collects information about the information system or host, with the intention of outsmarting the security controls.

[1] Jeya P. G., Ravichandran, M., Ravichandran, C. S.(2012). "Efficient Classifier for R2L and U2R Attacks". International Journal of Computer Applications, Volume 45, No.21. (retrieved 20 April 2019).

3. DATA DICTIONARY

The data schema of the KDD 1999 dataset is listed below as provided by KDD and the University of California in the data repository. A good explanation was adapted from the definition of the GureKddCup Dataset[2] and KDD[3].

3.1. Intrinsic attributes

FEATURE NAME	DESCRIPTION	TYPE
duration	length (number of seconds) of the connection	integer
protocol_type	type of the protocol, e.g. tcp, udp, etc.	factor
service	network service on the destination, e.g., http, telnet, etc.	factor
src_bytes	number of data bytes from source to destination	integer
dst_bytes	number of data bytes from destination to source	integer
flag	normal or error status of the connection	factor
land	1 if connection is from/to the same host/port; 0 otherwise	factor
wrong_fragment	number of ``wrong'' fragments	Integer
urgent	number of urgent packets	integer

Table 1 - Basic features of individual TCP connections.

3.2. Content Attributes

FEATURE NAME	DESCRIPTION	TYPE
hot	number of ``hot'' indicators	Integer
num_failed_logins	number of failed login attempts	Integer
logged_in	1 if successfully logged in; 0 otherwise	Factor
num_compromised	number of ``compromised'' conditions	Integer
root_shell	1 if root shell is obtained; 0 otherwise	Factor
su_attempted	1 if ``su root'' command attempted; 0 otherwise	Factor
num_root	number of ``root'' accesses	Integer
num_file_creations	number of file creation operations	Integer
num_shells	number of shell prompts	Integer
num_access_files	number of operations on access control files	Integer
num_outbound_cmds	number of outbound commands in an ftp session	Integer
is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise	factor
is_guest_login	1 if the login is a ``guest''login; 0 otherwise	factor

Table 2 - Content features within a connection suggested by domain knowledge.

---

[2] I. Perona, I. Gurrutxaga, O. Arbelaitz, J.I. Martín, J. Muguerza, J.M. Pérez. *"Service-independent payload analysis to improve intrusion detection in network traffic".* Proceedings of the 7th Australasian Data Mining Conference (AusDM08), Adelaide, Australia, 171-178, 2008. (retrieved 15 April 2019).

[3] UCI Knowledge Discovery in Databases Archive. (retrieved 15 April 2019). Available at https://kdd.ics.uci.edu/databases/kddcup99/task.html

3.3. Time Traffic Attributes

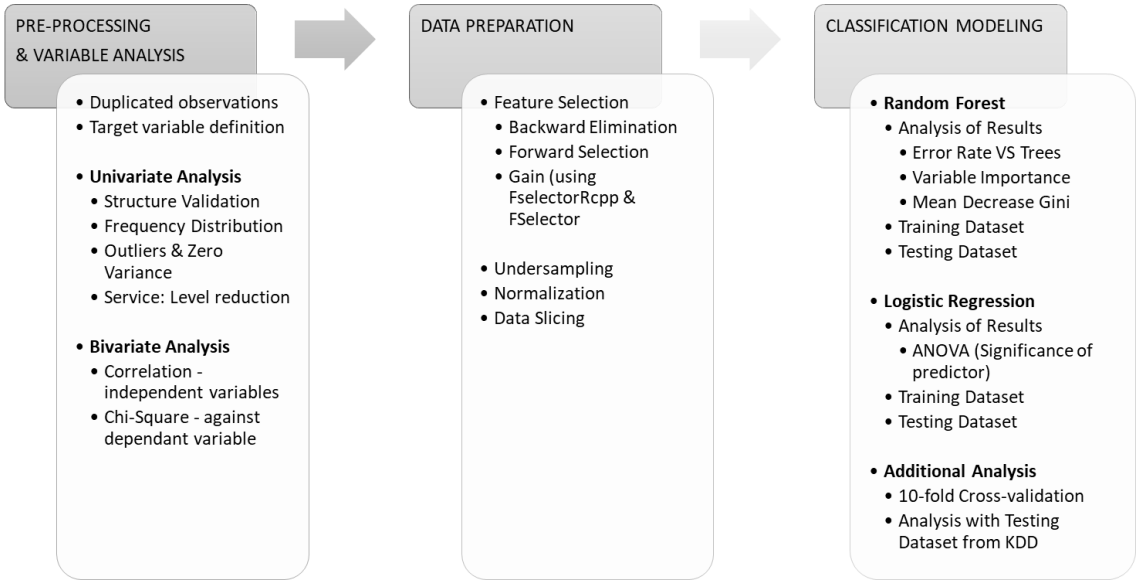
FEATURE NAME	DESCRIPTION	TYPE
count	number of connections to the same host as the current connection in the past two seconds	integer
serror_rate	% of connections that have ``SYN'' errors (same-host connection)	Integer
rerror_rate	% of connections that have ``REJ'' errors (same-host connection)	numeric
same_srv_rate	% of connections to the same service (same-host connection)	numeric
diff_srv_rate	% of connections to different services (same-host connection)	numeric
srv_count	number of connections to the same service as the current connection in the past two seconds (same host connection)	numeric
srv_serror_rate	% of connections that have ``SYN'' errors (same-service connection)	numeric
srv_rerror_rate	% of connections that have ``REJ'' errors (same-service connection)	numeric
srv_diff_host_rate	% of connections to different hosts (same-service connection)	numeric

Table 3 - Traffic features computed using a two-second time window.

3.4. Machine Traffic Attributes

FEATURE NAME	DESCRIPTION	TYPE
dst_host_count	sum of connections to the same destination IP address	integer
dst_host_srv_count	sum of connections to the same destination port number	integer
dst_host_same_srv_rate	the percentage of connections that were to the same service, among the connections aggregated in dst_host_count (32)	numeric
dst_host_diff_srv_rate	the percentage of connections that were to different services, among the connections aggregated in dst_host_count (32)	numeric
dst_host_same_src_port_rate	the percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count (33)	numeric
dst_host_srv_diff_host_rate	the percentage of connections that were to different destination machines, among the connections aggregated in dst_host_srv_count 33)	numeric
dst_host_serror_rate	the percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_count (32)	numeric
dst_host_srv_serror_rate	the percent of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_srv_count (33)	numeric
dst_host_rerror_rate	the percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_count (32)	numeric
dst_host_srv_rerror_rate	the percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_srv_count (33)	numeric

4. ANALYSIS APPROACH



5. PRE-PROCESSING

5.1. Duplicated observations

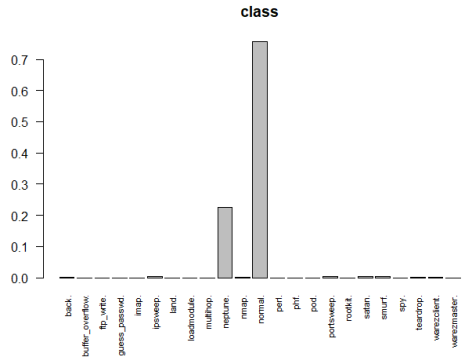
As it was mentioned above, the original dataset is highly imbalanced leaning towards the normal connections, and the percentage of duplicated records is more than 78%. The original dataset has 4,898,431 observations and new dataset has only 1,074,992.

Imbalance on data and duplicate records, can affect dramatically the performance of the classification model, that is why the first step of the data cleaning had to be executed.

Apache Hive was used during the first step of processing of the dataset. The code is available in the Capstone Project Repository on Github for further analysis.

5.2. Target variable definition

The original dataset contains 22 different types of attacks, and the scope of work for this project was defined as a binary classification model for intrusion systems, they were merged in a single class marked as “TRUE”, on the other hand, the normal traffic was label as “FALSE”



6. UNIVARIATE ANALYSIS

As part of the data cleaning process the following tasks were executed:

6.1. Structure of the dataset

Summary and Structure functions were used to explore how the dataset was built, correspondence between the attributes and the values, in accordance with the data dictionary.

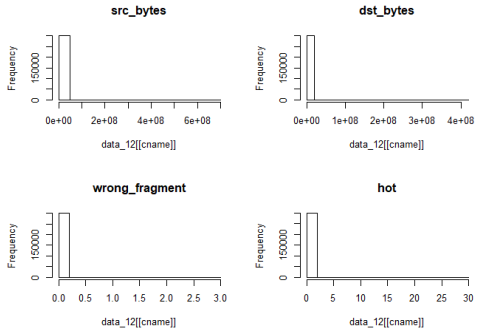
The attributes land, logged in, root_shell, su_attempted, is_host_login, and is_guest_login were converted to factors since they are categorical variables.

6.2. Frequency Distribution

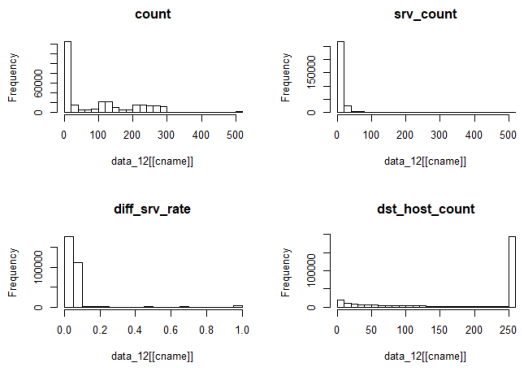
Histograms and boxplot were performed for numeric attributes, and barplot for non-numeric, the following was found:

Histogram:

- The normality test is not required, the histogram reflects that they are not following a theoretical normal distribution.
- When analyzing the data distribution for numerical attributes, it is easy to confirm that for most of them, data is lying in value zero (0), that shows a high imbalance on the attributes.



- For the attributes labeled as rate, data lies mostly in the values zero (0) and one (1).

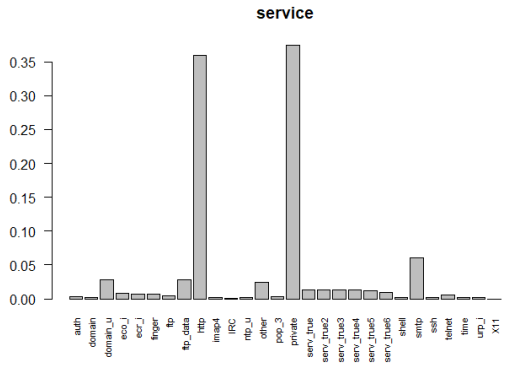


Boxplot:

- As seen in the histogram, for most of the attributes, data is mainly concentrated in zero (0) value.
- Mean, median, and quartiles are not divisible, all of them are shown very close to where the data is concentrated.
- For attributes as dst_bytes and wrong fragment, there are noticeable data points, that could be evaluated as outliers. Further analysis will demonstrate if they are, and if they could be removed.

Barplot:

- The most frequent protocol type is tcp and the most common service is http.

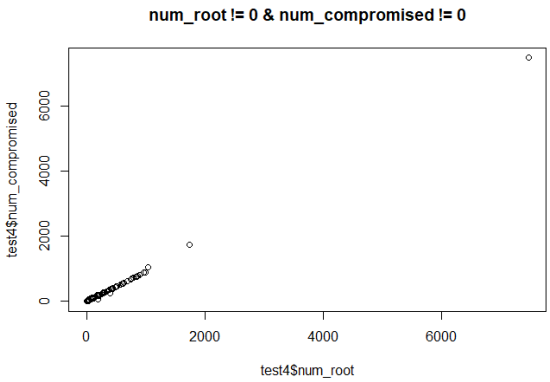


- Attributes land, root_shell, su_attempted, is_host_login, is_guest_login in more than 98% are catalogued as 0.
- For the original class attribute, more than70% of the data is cataloged as normal, 20% is classified as Neptune attack, that is a clear unbalance on the data.

6.3. Outliers processing

Outliers were detected for the attributes: duration, hot, num_root, src_bytes and dst_bytes.

As most of the data is concentrated in value zero (0), as per the mathematical approach the outliers will be most of the data that is different than value zero (0), that's why a further analysis was required in this case.



For duration, hot and num_root, only the extreme outliers were deleted manually after confirming they were classified as normal traffic, and since de data is unbalanced towards that category, the deletion of a few records won't affect the performance in a big proportion.

For src-bytes and dst_bytes, the outliers were not deleted because their class attribute value is TRUE, and its deletion could affect the representation of this class as part of the model.

6.4. Zero Variance and Near Zero Variance

The analysis of near zero variance will be useful during Dimensionality reduction process, there are various attributes that are considered Near to Zero Variance that potentially won't contribute significantly to the classification algorithm.

This process will be particularly important when randomly subsampling the dataset, due to the imbalance in some of the variables, potentially the underrepresented values won't be picked, leading to zero variance error in the classification algorithms

#	NZV_attribute	zeroVar	nzv
1	duration	FALSE	TRUE
2	src_bytes	FALSE	TRUE
3	land	FALSE	TRUE
4	wrong_fragment	FALSE	TRUE
5	urgent	FALSE	TRUE
6	hot	FALSE	TRUE
7	num_failed_logins	FALSE	TRUE
8	num_compromised	FALSE	TRUE
9	root_shell	FALSE	TRUE
10	su_attempted	FALSE	TRUE
11	num_root	FALSE	TRUE
12	num_file_creations	FALSE	TRUE
13	num_shells	FALSE	TRUE

14	num_access_files	FALSE	TRUE
15	num_outbound_cmds	TRUE	TRUE
16	is_host_login	FALSE	TRUE
17	is_guest_login	FALSE	TRUE
18	same_srv_rate	FALSE	TRUE
19	dst_host_count	FALSE	TRUE

The attribute num_outbound_cmds, was categorized as zeroVar, that confirms that it can be eliminated from the dataset that is going to be used for modeling.

On the other hand, the attribute land, can be decisive in the classification of land attacks, when land = 1; the positive attacks are catalogued as type land in more than 80% of the cases, when comparing against the original data set target attribute. In this case, that variable cannot be dropped.

6.5. Level reduction for Attribute Service

This attribute contain 70 levels (70 type of services), that for future, will be a problem for different algorithms (random forest can process up to 53 levels).

Analyzing the distribution of each service regarding the new_class attribute, there are some similarities, for various of them, the whole distribution leans towards TRUE.

As a method of reduction of levels, the types of services that have 900 or more observations, and all of them are valued as TRUE, will be renamed as a new service called “serv_true”, in total, 6 new levels were created to consolidate 39 service categories.

7. BIVARIATE ANALYSIS

7.1. Correlation Analysis Numeric attributes

There are various attributes that are correlated to each other, most of them are rates, which are attributes calculated from others. It is not recommended to delete them without having specific knowledge on the field.

During the process of dimensionality reduction, the correlated attributes will be evaluated and discarded as needed.

In order to verify the high positive correlation between the variables that are predominantly value = 0, two new subsets were created, and the conclusion was that there is a true correlation between the variables since the majority of the results (value = 0) were discarded and still the correlation looks the same.

For further analysis, the assumption is that behavior will be similar for the other variables that are highly correlated, further analysis will be executed when the features for the algorithm are selected.

The same concept was applied to the negative correlation, in this case, same_srv_rate was described as predominantly = 1, and dst_host_srv_serror_rate is predominantly = 0, discarding those values, we could notice that the correlation between the remaining results is only -0.32, showing that when the values 0 and 1 were deleted from the analysis, there is not strong correlation between variables.

7.2. Chi-Square

In order to determine which of the attributes are dependent to the class attribute "new_class". When chi-square is high, and p-value is low than alpha = 0.05 looking at the results, indicates that both variables are dependent to each other.

- "urgent" is the only numeric variable that is not dependent to the class attribute "new_class".

```
##                                  attribute Chi-squared value p-value
## 1                    train_label_5$duration              29836       0
## 2                   train_label_5$src_bytes             885469       0
## 3                   train_label_5$dst_bytes             700463       0
## 4               train_label_5$wrong_fragment               3479       0
## 5                       train_label_5$urgent                  1  0.9633
## 6                          train_label_5$hot               3355       0
## 7            train_label_5$num_failed_logins                 42       0
## 8              train_label_5$num_compromised               2006       0
## 9                     train_label_5$num_root               1809       0
## 10         train_label_5$num_file_creations                692       0
## 11                 train_label_5$num_shells                105       0
## 12           train_label_5$num_access_files               1368       0
## 13                      train_label_5$count             904867       0
## 14                  train_label_5$srv_count             125310       0
## 15                train_label_5$serror_rate             779085       0
## 16            train_label_5$srv_serror_rate             776297       0
## 17                train_label_5$rerror_rate              54237       0
## 18            train_label_5$srv_rerror_rate              50521       0
## 19              train_label_5$same_srv_rate             976322       0
## 20              train_label_5$diff_srv_rate             970907       0
## 21         train_label_5$srv_diff_host_rate             131120       0
## 22             train_label_5$dst_host_count             323152       0
## 23         train_label_5$dst_host_srv_count             779945       0
## 24      train_label_5$dst_host_same_srv_rate             823739       0
## 25      train_label_5$dst_host_diff_srv_rate             778917       0
## 26 train_label_5$dst_host_same_src_port_rate             313237       0
## 27 train_label_5$dst_host_srv_diff_host_rate             278437       0
## 28        train_label_5$dst_host_serror_rate             780366       0
## 29    train_label_5$dst_host_srv_serror_rate             778280       0
## 30        train_label_5$dst_host_rerror_rate              54687       0
## 31    train_label_5$dst_host_srv_rerror_rate              99285       0
```

- "is_host_login" is the only non-numeric attribute considered non-dependent.

```
##                    attribute Chi-squared value p-value
## 1   train_nonum$protocol_type              30684       0
## 2         train_nonum$service             960182       0
## 3            train_nonum$flag             900093       0
## 4            train_nonum$land                 33       0
## 5       train_nonum$logged_in             576410       0
## 6      train_nonum$root_shell                 37       0
## 7    train_nonum$su_attempted                 33       0
## 8   train_nonum$is_host_login                  1  0.4219
## 9  train_nonum$is_guest_login                619       0
## 10          train_nonum$class            1074986       0
```

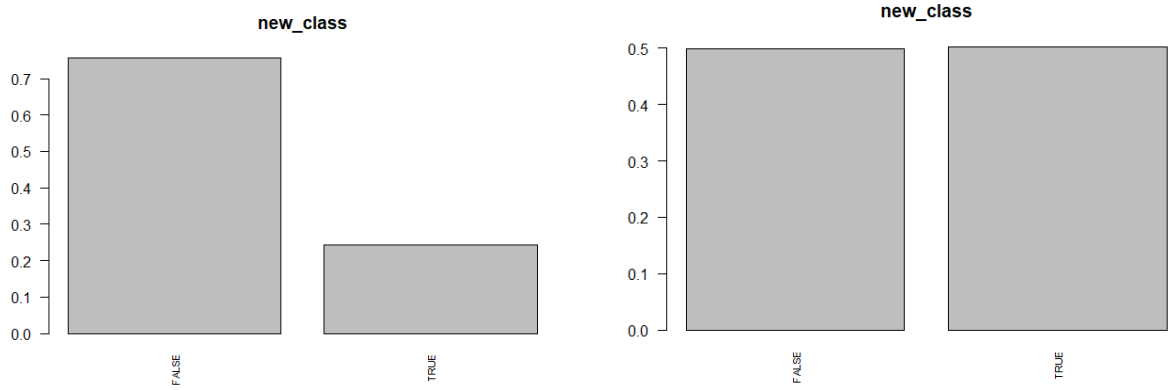
8. DATA PREPARATION FOR MODELING

8.1. Feature Selection

- **Backward Elimination:** The goal of this method is to build a multiple regression model that includes as few attributes as possible. The result of this regression suggested in this case that the attribute dst_host_name_srv_rate has to be eliminated.
- **Forward Selection:** After a long processing time, the method suggested the elimination of srv_diff_host_rate
- **Gain (Using FSelectorRcpp Function):** It is an entropy-based Feature Selection Algorithm based on multi-Interval Discretization. The result suggests the elimination of urgent (coincident with Chi-Square exploration) and num_failed_logins.
- **Gain (FSelector):** Similar to FSelectorRcpp, this algorithm suggested the deletion of duration and protocol type.

8.2. Undersampling

The dataset will be under sampled to 300,000 records, a random process will be used to balance the dependent attribute trying to have the same representation, it reduces the number of "FALSE" observations and while maintaining and compensating (if needed) the "TRUE" observations.



Class attribute must be converted to factor in order to process a sampling function.

8.3. Normalization

Normalization of numeric attributes is required before processing the model, this due to the diversity of ranges in the numeric attributes, some are listed from 0 to 1, others could go from 0 to 50,000. Performing this step, the possible biased impact of a variable due to a large numeric value is avoid on the prediction.

8.4. Data Slicing

For the purpose of this project, the test dataset will be randomly divided in two: 70% of the records will be training, 30% of data will be testing.

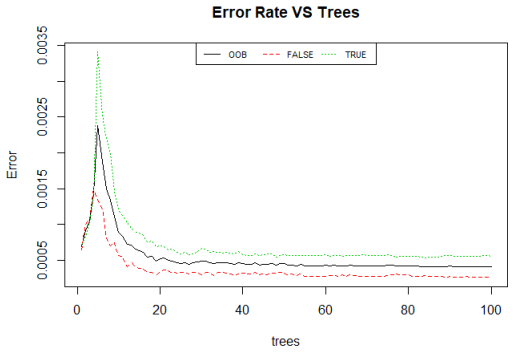
NOTE: Some repository directories for the KDD 99 Dataset provides a testing dataset, for the purpose of this project, it was not considered as part of the Scope.

9. CLASSIFICATION MODELING

9.1. Random Forest

The following results can be obtained from the Random Forest Function

- **Error Rate VS Trees:** The plot (error rate VS trees) explains that for the random forest algorithm for 15 or less tress, the level of randomness is high, leading to a higher error rate, after 20 tress the behaviour is virtually flat.



- **Variable Importance:** The attributes service, hot, src_bytes, and srv_count, were determined as best in ranking for this measure.

- **Mean Decrease Gini:** Service was also selected as the attribute with the highest variable importance. scr_bytes and flag, were catalogued as second and third, but with less Gini Index.

9.1.1. Confusion Matrix – Training Dataset

	FALSE	TRUE
FALSE	104615	18
TRUE	7	105360

RATE	RESULT
Accuracy	0.9999
Misclassification Rate	0.0001
True Positive Rate	0.9999
False Negative Rate	0.0001
True Negative Rate	0.9998
False Positive Rate	0.0002
Precision	0.9998
Prevalence	0.5017

Accuracy on the training dataset is high, surprisingly, the reduction on service classes, under sampling and new definition on class attribute did not impact the model definition.

9.1.2. Confusion Matrix – Testing Dataset

	FALSE	TRUE
FALSE	44827	26
TRUE	11	45136

RATE	RESULT
Accuracy	0.9996
Misclassification Rate	0.0004
True Positive Rate	0.9998
False Negative Rate	0.0002

True Negative Rate	0.9994
False Positive Rate	0.0006
Precision	0.9994
Prevalence	0.5015

Accuracy on the classification of the test dataset is 99.96%, this could be due the data is a subset of the original dataset and the distribution of the data is similar. For further analysis, this model could be replicated on the original test dataset provided by KDD.

9.2. Logistic Regression

The results of the logistic regression showed that all the attributes selected for this model are statistically significant.

9.2.1. Analysis of Variance

The p-value in this case, indicate the significance of the predictor variable on the probability of achiving a success (TRUE),

- scr_bytes, dst_bytes, dst_host_srv_diff_host_rate, logged_in, root_shell, and is_guest_login are not considered significant in the prediction of TRUE values in the class attribute.
- dst_host_srv_diff_host_rate actually increases the residual deviance, for future the variable can be dropped to see the difference
- service and flag provide the more drop on the residual deviance while being statistically significant
- A large p-value indicates that the model without the variable explains more or less the same amount of variation
- The intention here is to see a drop-in deviance and AIC

9.2.2. Confusion Matrix – Training Dataset

	FALSE	**TRUE**
FALSE	103788	974
TRUE	834	104404

RATE	**RESULT**
Accuracy	0.9914
Misclassification Rate	0.0086
True Positive Rate	0.9921
False Negative Rate	0.0079
True Negative Rate	0.9907
False Positive Rate	0.0093
Precision	0.9908
Prevalence	0.4972

9.2.3. Confusion Matrix – Testing Dataset

	FALSE	TRUE
FALSE	44467	401
TRUE	371	44761

RATE	RESULT
Accuracy	0.9914
Misclassification Rate	0.0086
True Positive Rate	0.9918
False Negative Rate	0.0082
True Negative Rate	0.9911
False Positive Rate	0.0089
Precision	0.9911
Prevalence	0.4973

9.2.4. Confusion Matrix – Testing Dataset

As part of the analysis of the performance of the model, a Logistic Regression using 10-fold Cross Validation has been executed, the accuracy is low (0.991%), but like the other models.

10. ADDITIONAL ANALYSIS

Test with 10-fold cross validation

A 10-fold cross validation analysis will be performed, to confirm the performance of the Random Forest Classifier. The goal is to test the model capacity to predict new data that was not used to estimate it, also to identify problems with overfitting or biased selection (non-random selection), a loop will run 10 times, creating new training and test dataset, the accuracy will be measure for every iteration. The results will confirm the performance of the prediction model.

Test using Test Dataset (KDD)

As mentioned early in this chapter, the Dataset provided by KDD in their repository will be used to confirm the performance of the model created using only Random Forest, which was the one that scored the better accuracy

	FALSE	TRUE
FALSE	47031	4412
TRUE	882	24966

RATE	RESULT
Accuracy	0.9315
Misclassification Rate	0.0685
True Positive Rate	0.9659
False Negative Rate	0.0341
True Negative Rate	0.9142
False Positive Rate	0.0858
Precision	0.8498
Prevalence	0.3230

The result of the prediction against the new data set is (93%), considered very good, since the data was extracted differently and was mentioned in various research papers that was distributed differently.

11. RECOMMENDATIONS

- Perform the part of the processing required for this project in a distributed type of environment, 4MM records x 43 attributes will be difficult to be handled by R in a consumer type of computer. Data cleaning and data preparation stages performed better in Hive that in R.
- Find the proper methodology that will allow executing processes like Variable Independence and Feature Selection to be developed in parallel (another script file in the project), and only the results are to be leveraged as part of the core piece of programming. Rely on one document for every calculation demands high processing resources and a long waiting time in processing while generating the partial reports.
- Analyze further the relationship between categorical attributes, not only each of them against the class attribute rather each of the independent attributes against each other.
- Find a better method to analyze the impact of each of the different levels part of the "Service" attribute, looking for an alternative to scale down the number of services evaluated. Domain Knowledge in this field might be required.
- Investigate further on the behavior of the variables like "wrong fragment" and "num_access_files" and their outliers, implementing a method of imputation like to K-Nearest Neighbors.
- Investigate further in the appropriated methodology that will allow transforming the numeric attributes to reduce partially the concentration of records to value zero (0). Example, Log Transformation.
- Implement a Cross-Validation across the modeling process, reducing the oversampling on the datasets.

12. FUTURE WORK

For future work on this project, the following tasks can be executed

- Reduce the number of attributes to a maximum of 15, in order to compare the results against the 28 attributes chosen for this project. Example: Near-Zero Variance Analysis suggested the deletion of various attributes, that could be a good point of start
- Since the scope for this project was narrowed to be a binary classification problem, it will be interesting to create a model that is able to classify within the 22 types of different attacks.
- Make a comparison between the performance of the dataset that contains duplicates (raw data) and the dataset used for this project.
- Perform the modeling process on a dataset that hasn't been subsampled and compares the results with the present document.

- Produce curves and calculations like ROC (Receiver Operating Characteristics) and AUC (Area Under Curve) that will allow comparing better the performance between the Random Forest and Logistic Regression classification models.
- Perform Principal Component Analysis and use resultant data for classification modeling.

13. CONCLUSIONS

- Results of accuracy in more than 90% allow to think that the model is overfitted, Further analysis of the dataset used for modeling will be required, techniques to prevent overfittings like early cross-validation, strict feature selection, outlier treatment, or regularization might be required to improve the classification model.
- "src_bytes" and "dst_bytes" are some of the more relevant attributes for the classification model.
- Analysis on numeric attributes demonstrated that data was concentrated in values close to zero, Near Zero Variance Algorithm suggested the deletion of 19 of the 42 attributes (45% of independent variables), this two facts evidence that set in its integrity was concentrated giving few spaces for analysis of outliers or univariate clustering for identification of patterns.
- Data processing using R instead of a distributed computing environment increases the processing time for tasks related to data analysis and data cleaning, find the right balance or look for integration methodologies will allow that further analysis could be executed in a reduced time frame and will allow optimization of computing resources.
- Further investigation in Intrusion Detection Systems will help interpret better the dataset used as part of this document, type of service and type of flags could be consolidated in fewer levels, allowing the model to perform better.
- Domain knowledge will help to understand better the impact of the different rates included as part of the dataset, giving the chance to add value to the additional processing time required when dealing with 43 attributes in a dataset of 4 million records.
- Random Forest algorithm performed slightly better than logistic regression for this classification problem, but as the results for accuracy were very high, there is no preference for any of them.