Network Intrusion Detection System

Muskan Agrawal
(18BCE0707)
Syed Ayaz Imam
(18BCE0660)

A report Submitted for the J component of

CSE3502- Information Security Management

Supervisor: Dr.D.RUBY

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
Vellore institute of Technology, Vellore

June 1, 2021

Declaration

This report has been prepared on the basis of our own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 4306

Student Name:

Muskan Agrawal

Syed Ayaz Imam

Date of Submission:

1 June,2021

Signature:

TABLE OF CONTENTS

1.PROBLEM STATEMENT

1.1 Idea

As everyone wants a 100percent stable and secure solution to their network vulnerabilities. Despite extensive study into Intrusion Detection Systems and a plethora of antivirus software, there are currently no effective solutions until all of the parameters of an intrusion detection system are met, which are time consuming, cumbersome, and require constant updating. This project discusses the challenges that IDS developers confront in establishing an effective intrusion detection system, as well as its limitations and difficulties related to the development and organisation of intrusion detection systems.

1.2 Scope

The issue arises as to why, despite spending so much money on it, we are still unable to develop an intrusion detection system capable of preventing such attacks and losses. As they are based on abuse intrusion detection methods, antivirus systems cannot give the appropriate level of protection. These antivirus systems will not be able to cope with intruders' innovative and sophisticated ways unless they are identified. Anomaly Intrusion Detection Systems were created to solve this problem, and they can identify any unwanted change in network data or a divergence from regular data standards on the network, which implies they can detect novel intrusion types. However, the IDS has several deep flaws:

Problem 1: Software-based security systems are unable to provide adequate protection against new forms of threats.

Problem 2: After a length of time, the observed environment's behaviour may change, necessitating system retraining.

Problem 3: If malicious behaviour is present in the training set, the system will treat it as normal.
Problem 4: False Positives and False Negatives

To address these issues, the research work has focused on the following objectives:

Objective 1: The intrusion detection system (IDS) is used to detect anomalies using machine learning techniques like support vector machine (SVM) and random forest (RF).

Objective 2: Both the support vector machine and the random forest approaches are compared for accuracy and sensitivity.

Objective 3: Using a variety of test cases, verify and confirm the suggested methodology's outcomes.

1.3 Novelty

Our research examines the inherent flaw in the KDDcup 99 dataset and proposes a solution in the form of a study of the NSL-KDD dataset for determining intrusion detection accuracy. For the dataset, multiple classification algorithms with and without feature reduction were used in the experiment.

1.4 Comparative Study

S. No.	Paper Name	Year	Authors	Focus Area
1	Exploration of Hardware Architectures for String Matching Algorithms in Network Intrusion Detection Systems	2020	Rashid, M., Imran, M., & Jafri, A. R	The objective of this paper is to explore the performance of existing string matching algorithms, either implemented on FPGA or CMOS.

2	Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model	2018	Aljawarneh, S., Aldwairi, M., & Yassein, M. B.	The paper proposes a hybrid model for dimensionality reduction that improves the accuracy rate and reduces the detection time. The analysis performed on the NSL-KDD dataset through the help of tables and figures has allowed the researcher to gain a clearer dataset understanding.
3	Anomaly-based intrusion detection system	2019	Jyothsna, V., & Prasad, K. M	The objective of this paper is to study the NSL-KDD dataset and propose to optimize the features toward designing the scale to detect the intrusions.

4	Deep learning approach on network intrusion detection system using NSL-KDD dataset.	2019	Gurung, S., Ghose, M. K., & Subedi, A	The study proposes a solution to the NSL-KDD problem and develops a system for detecting such incursions. The system may be installed on any server and watches any organization's network activity in real time. The deep-net can detect any incursion and modify its classification based on recent data. Variations on the encoders can be made to make the system more

				robust and improve detection accuracy.
5	BAT: deep learning methods on network intrusion detection using NSL-KDD	2020	Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y	The authors propose a novel model BAT-MC via the two phase's learning of BLSTM and attention on the time series features for intrusion detection using NSL-KDD dataset. BLSTM layer which connects the forward LSTM and the backward LSTM is used to extract features on the traffic bytes of each packet.

Paper1:

TITLE, AUTHOR, AND JOURNAL:
Rashid, M., Imran, M., & Jafri, A. R. (2020, July). Exploration of Hardware Architectures for String Matching Algorithms in Network Intrusion Detection Systems. In *Proceedings of the 11th International Conference on Advances in Information Technology* (pp. 1-7).

OBJECTIVE:
Identification of frequently implemented string matching algorithms and techniques for NIDS.

METHODOLOGY USED:
The most commonly used string matching algorithms and strategies for hardware implementation are identified in this study. Following that, an examination of possible hardware architectures for the indicated algorithms and methodologies is offered. Finally, the implementation specifics of the investigated designs are

examined in terms of the device used, the hardware resources consumed, the operational clock frequency, and the throughput.

LIMITATION:
The large memory need for storing the transition rules of the underlying deterministic finite automaton is the limitation.

Paper2:

TITLE, AUTHOR, AND JOURNAL:
Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152-160.

OBJECTIVE:
Hybrid classification-based intrusion detection model

METHODOLOGY USED:
To begin, data must be filtered using the Vote algorithm with Information Gain, which combines the probability distributions of these basic learners to select the relevant attributes that have a favourable impact on the proposed model's accuracy. J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump, and NaiveBayes are the next classifiers in the hybrid approach. We see enhanced accuracy, a high false negative rate, and a low false positive rule based on the findings obtained utilising the proposed model.

LIMITATION:
Researchers should investigate the prospect of employing optimization approaches to develop an intrusion detection model with a higher accuracy rate.

Paper3:

TITLE, AUTHOR, AND JOURNAL:
Jyothsna, V. V. R. P. V., Prasad, R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, *28*(7), 26-35.

OBJECTIVE:
Summarization study and identification of the drawbacks of formerly surveyed works.

METHODOLOGY USED: This paper explains the fundamentals of the most common anomaly-based network intrusion detection systems, as well as their operational designs, and offers a classification system based on the type of processing that is related to the target system's "behavioural" model. In a succinct manner, this study also outlines the main features of different ID systems/platforms that are now available. The most important open concerns in Anomaly-based Network Intrusion Detection systems are identified, with a special emphasis on assessment.

LIMITATION: To deal with the ever-increasing attacks, faster and more effective countermeasures are required. We discovered that the majority of the works surveyed do not match these criteria.

Paper4:

TITLE, AUTHOR, AND JOURNAL:
Gurung, S., Ghose, M. K., & Subedi, A. (2019). Deep learning approach on network intrusion detection system using NSL-KDD dataset. *International Journal of Computer Network and Information Security*, *11*(3), 8-14.

OBJECTIVE:
Apply and Test Deep Learning Techniques to Develop Network Intrusion Detection Systems

METHODOLOGY USED:
The purpose of this research is to develop an intrusion detection system that not only learns but also adjusts to previously unknown patterns. A sparse auto-encoder was employed for unsupervised feature learning. A logistic classifier is then used to classify the NSL-KDD dataset. The accuracy, precision, and recall of the system have all been tested, with the findings proving to be very promising for future use and improvements.

LIMITATION:
The lack of a real-time pattern of network data containing both intrusions and routine uses, continually evolving and changing attack patterns, extensive training period, and insufficient knowledge about dataset updates are all limitations of this study.

Paper 5:

TITLE, AUTHOR, AND JOURNAL:
Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). BAT: deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access*, *8*, 29575-29585.

OBJECTIVE:
To solve the problems of low accuracy and feature engineering in intrusion detection, a traffic anomaly detection model BAT is proposed.

METHODOLOGY USED:
BLSTM (Bidirectional Long Short-Term Memory) and attention mechanisms are combined in the BAT model. The attention mechanism is used to screen the network flow vector, which is made up of packet vectors created by the BLSTM

model and can be used to classify network traffic. In addition, to capture the local aspects of traffic data, we use numerous convolutional layers. We call the BAT model BAT-MC because it uses several convolutional layers to process data samples. The softmax classifier is used to classify network traffic. The suggested end-to-end model does not require any feature engineering expertise and can learn the hierarchy's important characteristics automatically. It can efficiently represent network traffic flow and improve the capacity to detect anomalies. We put our model to the test on a publicly available benchmark dataset, and the findings show that it outperforms existing comparison methods.

LIMITATION: There is no mathematical study that connects the parameters to the rates of convergence. If the number of function evaluations is low, accuracy may be limited.

1.5 Dataset

The NSL-KDD dataset was used to test the effectiveness and practicality of the proposed IDS system.

The KDDcup99 dataset has been updated. In comparison to the KDDcup99 dataset, the NSL-KDD dataset provides a few advantages. It has addressed some of the flaws of the KDDcup99, which is widely used as a benchmark for intrusion detection testing. NSL-KDD, like KDDcup99, has a training dataset of about 4,900,000 single connection vectors, each of which has 41 features and is classified as either normal or attack type, with exactly one attack type. For the reasons listed below, NSL-KDD has become a more popular dataset than KDD Cup 99 for intrusion detection.

The training set's redundant records are removed. To increase intrusion detection performance, duplicate records from the test set are deleted. The use of the NSL-KDD dataset for classification allows for a precise assessment of various learning strategies. The NSL-KDD dataset is cost-effective to employ for experiments since it has a sufficient number of examples in both the training and testing sets.

1.6 Test Beds

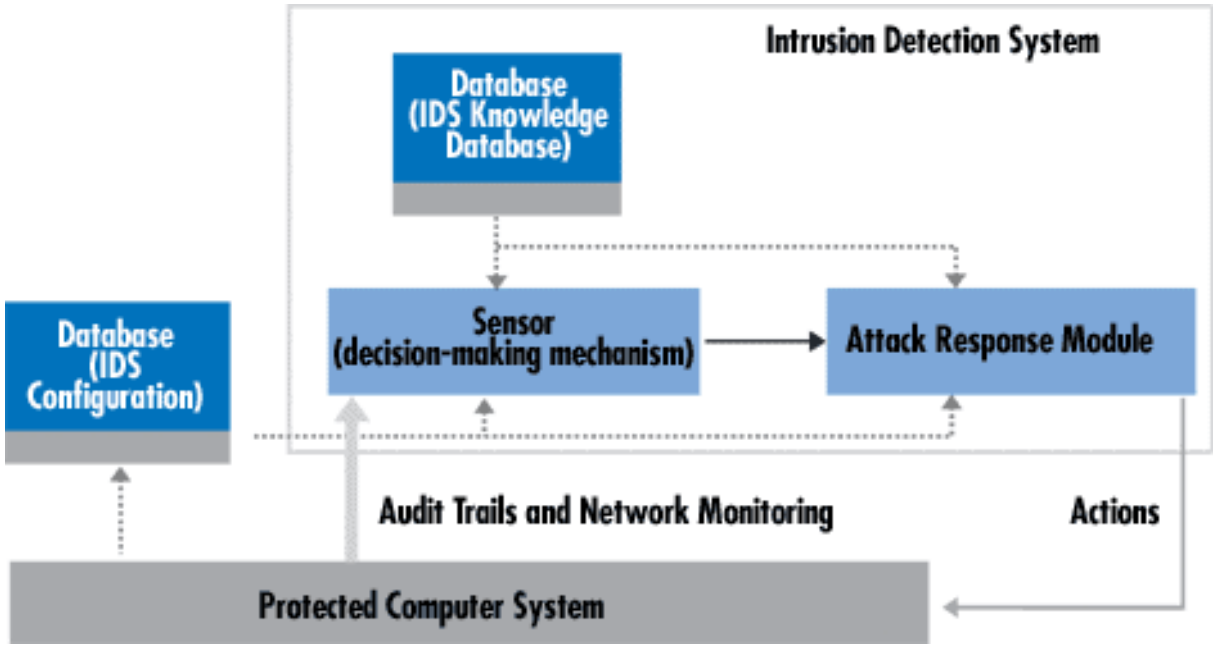
We built many machine learning models for classification in the proposed system. The model will recognize incursion activities while they are occurring. A classification task entails the use of training and testing sets, both of which contain instances. Each instance in the training set contains one “target value" (class labels: Normal or Attack) and several “attributes" (features). The model's purpose is to create one that predicts the target value of a data instance in a testing set with only attributes.

1.7 Expected Results

An Intrusion Detection System is essential for protecting a network from intrusion and ensuring the security of any data. The major goal is to improve the accuracy of a network intrusion detection system that adapts to different types of attacks.

2. ARCHITECTURAL DESIGN

The essential part of every intrusion detection system is a sensor (an analysis engine) that is responsible for identifying intrusions. This sensor has intrusion detection and decision-making mechanisms. Sensors get their raw data from three places: their own IDS knowledge base, syslog, and audit trails. The syslog may contain information such as file system settings, user authorizations, and so on. This data serves as the foundation for additional decision-making.



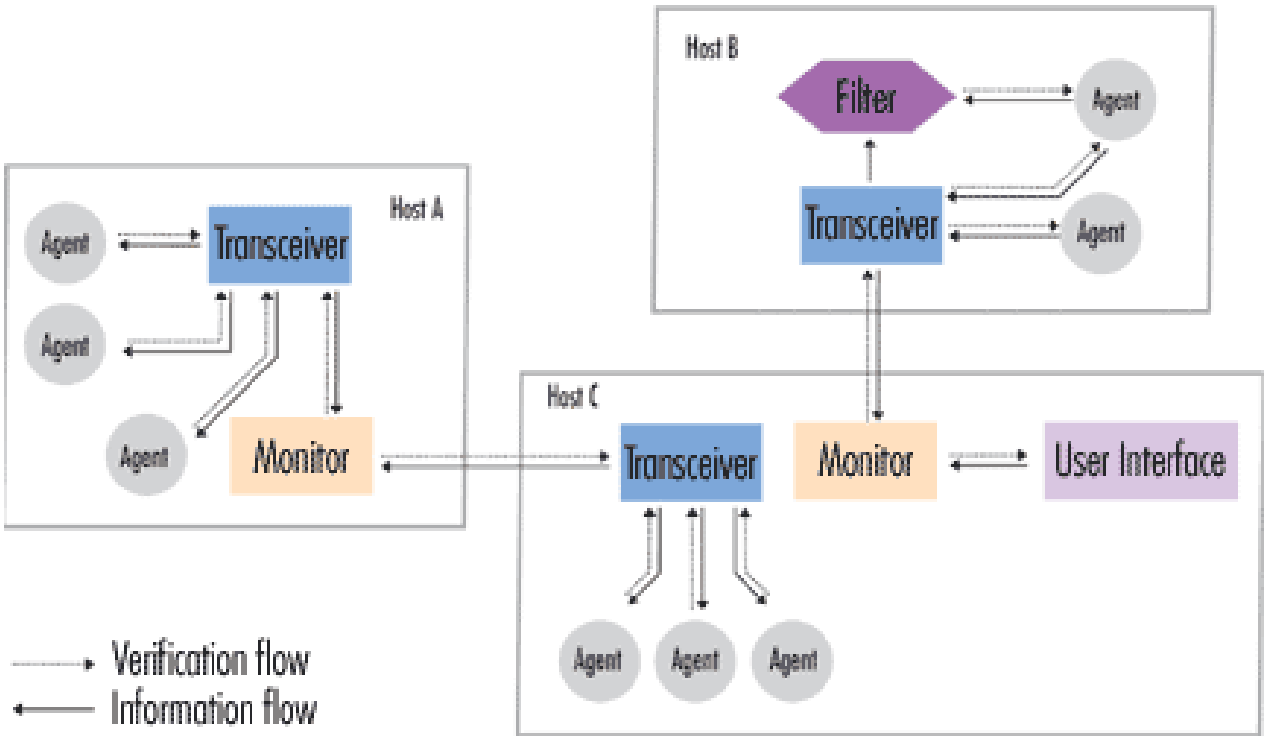
The sensor is connected to an event generator, which is in charge of data collecting.

The sensor's job is to filter information from the protected system's event set and eliminate any unnecessary data, resulting in the detection of suspicious behaviours. For this, the analyser consults the detection policy database. Attack signatures, typical behaviour profiles, and essential parameters are among the components of the latter (for example, thresholds). Intrusion detection systems can be centralised (physically integrated within a firewall, for example) or distributed.

A distributed IDS is made up of numerous Intrusion Detection Systems (IDS) that are joined via a broad network and communicate with one another. Small autonomous modules are grouped per-host throughout the protected network in more advanced systems, which follow the agent structure principle.
The agent's responsibility is to monitor and filter all activity within the protected region, perform an initial analysis, and perhaps take action based on the technique used.
Another aspect of the agent's function is its mobility and ability to move across various physical areas.

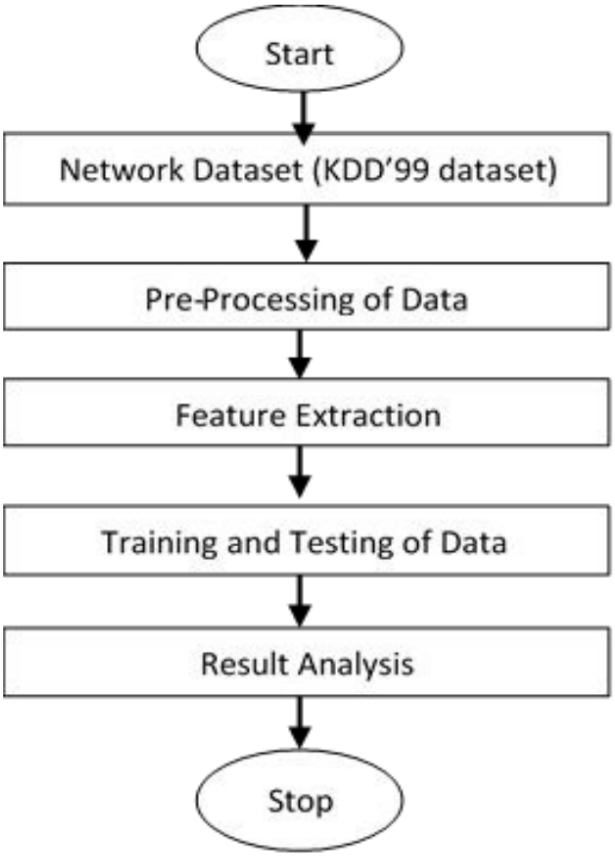
Aside from agents, the system may include transceivers to keep a record of all operations involving agents from a particular host.
The outcomes of transceivers' actions are always sent to a single display. Monitors gather data from a particular network region (rather than just a single

host), allowing them to correlate scattered data. Some filters for data selection and aggregation may also be introduced.



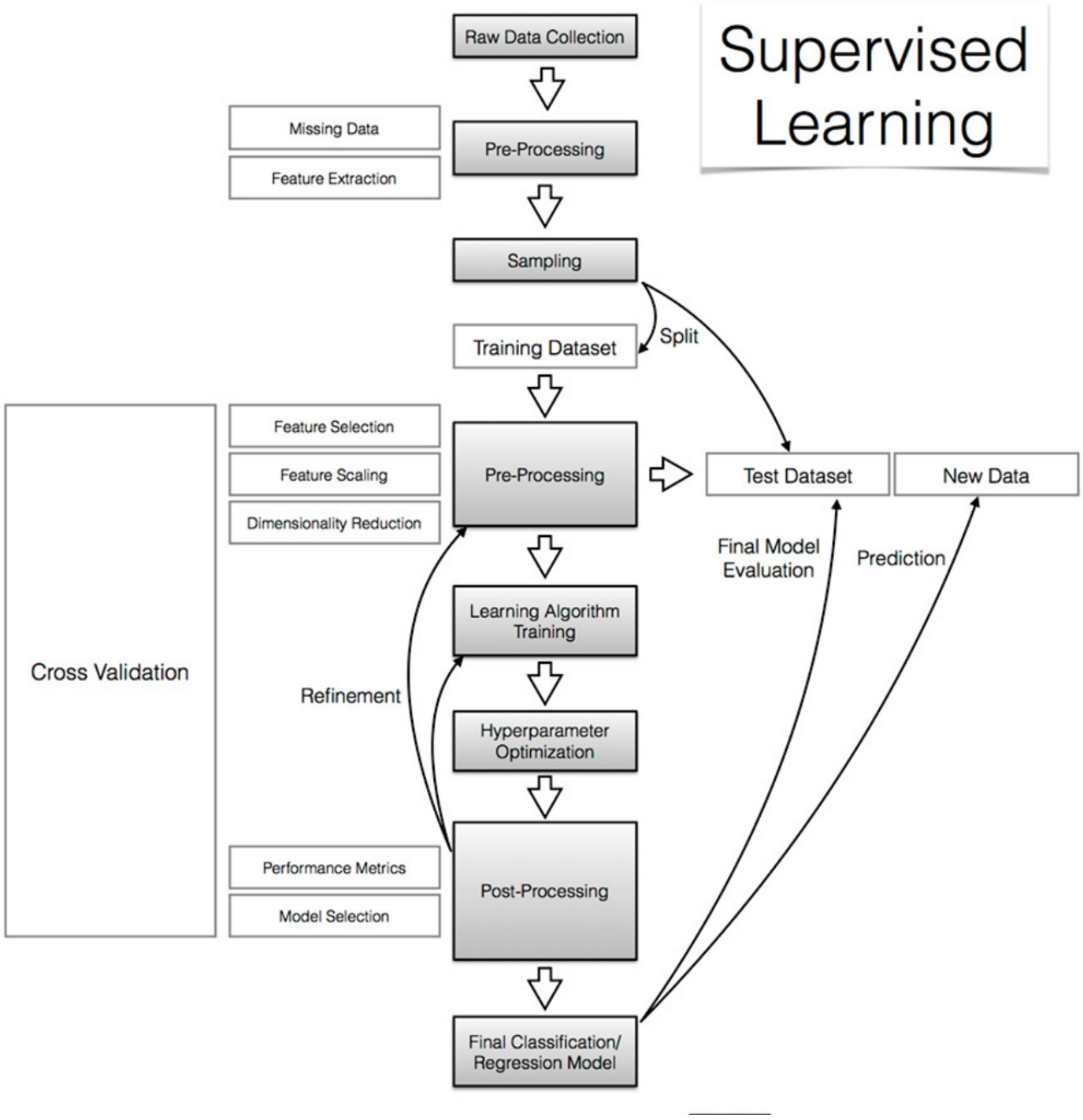
2.1 High Level Design

It senses and extrapolates knowledge using data processing techniques that can reduce the likelihood of fraud detection, increase audit reflexes to prospective business changes, and ensure that risks are managed in an extremely rapid and active manner. Internal auditors will choose from a variety of knowledge mining approaches in addition to using a specific data processing technology. Naive Bayes, decision trees, logistic regression, and closest neighbor methods are among the most commonly utilized techniques. Each methodology examines information in a different way.



2.1 Low Level Design

Machine learning is a technology that requires a large amount of data to train the model and predict future outcomes. When a model learns from data properly, it has a high chance of properly predicting the future. Machine learning techniques are typically employed when a problem cannot be solved only through mathematical calculations or script writing. There are two types of machine learning issues that can be dealt with. There are two types of learning: supervised and unsupervised.



Before training the algorithms in supervised learning, a preset dataset is provided. To begin, these datasets are labeled, and algorithms are learned based on the labels or tags. The model may anticipate any future expectations after learning from the dataset.

3. IMPLEMENTATION

3.1 MODULE DESCRIPTION

a) Feature Scaling

Feature scaling is a strategy for putting the data's independent features into a set range. It is used to handle significantly changing magnitudes, values, or units during data pre-processing. If feature scaling is not done, a machine learning algorithm will assume larger values to be higher and smaller values to be lower, regardless of the unit of measurement.

In [15]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

# extract numerical attributes and scale it to have zero mean and unit variance
cols = dfkdd_train.select_dtypes(include=['float64','int64']).columns
sc_train = scaler.fit_transform(dfkdd_train.select_dtypes(include=['float64','int64']))
sc_test = scaler.fit_transform(dfkdd_test.select_dtypes(include=['float64','int64']))

# turn the result back to a dataframe
sc_traindf = pd.DataFrame(sc_train, columns = cols)
sc_testdf = pd.DataFrame(sc_test, columns = cols)
```

b) Encoding

We frequently work with datasets in machine learning that include numerous labels in one or more columns. Labels in the form of words or numbers can be used. The training data is frequently labeled in words to make it intelligible or human readable.

The encoding method is straightforward, and it entails transforming each value in a column to a number. Consider a dataset of bridges with a column named bridge-types and values in the range below. Though there will be many more columns in the dataset, we will simply focus on one categorical column to explain label-encoding.

In [16]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()

# extract categorical attributes from both training and test sets
cattrain = dfkdd_train.select_dtypes(include=['object']).copy()
cattest = dfkdd_test.select_dtypes(include=['object']).copy()

traincat = cattrain.apply(encoder.fit_transform)
testcat = cattest.apply(encoder.fit_transform)

# separate target column from encoded data
enctrain = traincat.drop(['attack_class'], axis=1)
enctest = testcat.drop(['attack_class'], axis=1)

cat_Ytrain = traincat[['attack_class']].copy()
cat_Ytest = testcat[['attack_class']].copy()
```

c) Sampling

The reason seems to be that many machine learning algorithms are built to work with classification data in which each class has an equal number of observations. When this isn't the case, algorithms can learn that a small number of samples are irrelevant and may be ignored in order to produce good results.

Data sampling is a set of approaches for transforming a training dataset in order to balance or improve the distribution of classes. Standard machine learning methods can be trained directly on the altered dataset without any modifications once the dataset has been balanced. This enables a data preparation strategy to meet the difficulty of imbalanced classification, even with substantially imbalanced class distributions.
There are many various types of data sampling strategies that can be utilized, and there is no one-size-fits-all solution for all classification issues and models.

In [17]:
```
from imblearn.over_sampling import RandomOverSampler
from collections import Counter

# define columns and extract encoded train set for sampling
sc_traindf = dfkdd_train.select_dtypes(include=['float64','int64'])
refclasscol = pd.concat([sc_traindf, enctrain], axis=1).columns
refclass = np.concatenate((sc_train, enctrain.values), axis=1)
X = refclass

# reshape target column to 1D array shape
c, r = cat_Ytest.values.shape
y_test = cat_Ytest.values.reshape(c,)

c, r = cat_Ytrain.values.shape
y = cat_Ytrain.values.reshape(c,)

# apply the random over-sampling
ros = RandomOverSampler(random_state=42)
X_res, y_res = ros.fit_sample(X, y)
print('Original dataset shape {}'.format(Counter(y)))
print('Resampled dataset shape {}'.format(Counter(y_res)))
```

```
Original dataset shape Counter({1: 67343, 0: 45927, 2: 11656, 3: 995, 4: 52})
Resampled dataset shape Counter({1: 67343, 0: 67343, 3: 67343, 2: 67343, 4: 67343})
```

d) Feature Selection

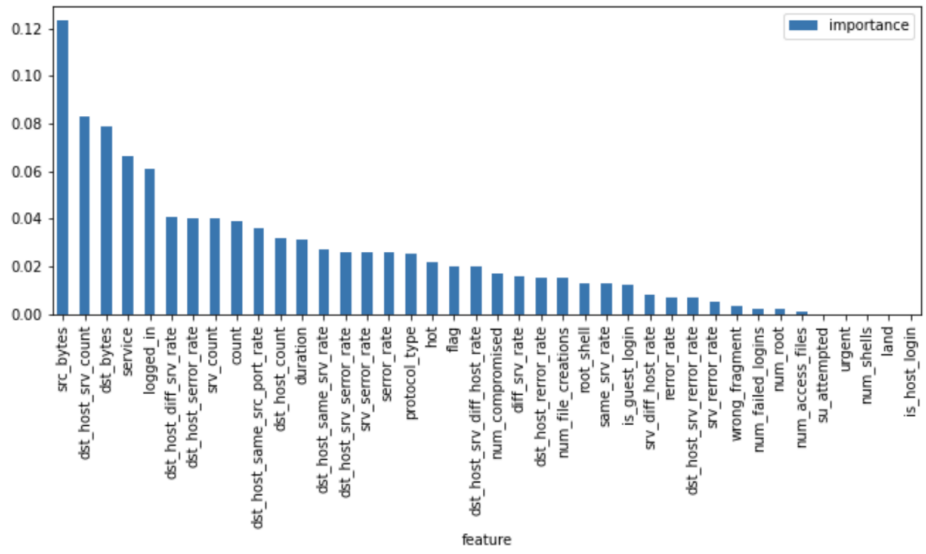
When creating a predictive model, feature selection is the process of minimizing the number of input variables.

The number of input variables should be reduced to lower the computational cost of modeling and, in some situations, to increase the model's performance.

The relationship between each input variable and the goal variable is evaluated using statistics, and the input variables with the strongest link with the target variable are selected. Although the choice of statistical measures depends on the data type of both the input and output variables, these methods can be quick and successful.

In [18]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier();

# fit random forest classifier on the training set
rfc.fit(X_res, y_res);
# extract important features
score = np.round(rfc.feature_importances_,3)
importances = pd.DataFrame({'feature':refclasscol,'importance':score})
importances = importances.sort_values('importance',ascending=False).set_index('feature')
# plot importances
plt.rcParams['figure.figsize'] = (11, 4)
importances.plot.bar();
```



3.2 ALGORITHMS USED

a) Logistic Regression

The logistic regression which is at the heart of the procedure, is called logistic regression.
The logistic function, also known as the sigmoid function, was created by statisticians to characterize the characteristics of population increase in ecology, such as how it rises swiftly and eventually reaches the environment's carrying capacity. It's an S-shaped curve that can transfer any real-valued integer to a value between 0 and 1, but never exactly between those two points.
Logistic regression, like linear regression, uses an equation as its representation.

In order to forecast an output value, input data (x) are linearly combined with weights or coefficient values (abbreviated as Beta in Greek) (y). A basic difference from linear regression is that the output value being modeled is a binary value (0 or 1) instead of a numeric value.

b) Naive-Bayes

It's a classification method based on Bayes' Theorem and the assumption of predictor independence. A Naive Bayes classifier, in simple terms, posits that the existence of one feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is simple to construct and is especially good for huge data sets. Naive Bayes is renowned to outperform even the most advanced classification systems due to its simplicity.

c) Decision Tree

A decision tree is a flowchart-like structure in which each node represents a "test" on an attribute (for example, determining whether a coin flip will come up heads or tails), each fork represents the test's conclusion, and each leaf node provides a class label (decision taken after computing all attributes).

The paths from root to leaf describe the categorization rules.

In decision analysis, a decision tree and its closely related impact diagram are used to calculate the expected values (or expected utility) of competing alternatives using a visual and analytical decision assistance tool.

There are three sorts of nodes in a decision tree:

- Squares are commonly used to symbolise decision nodes.
- Chance nodes are usually depicted as circles.
- End nodes - triangles are commonly used to depict them.

d) KNN

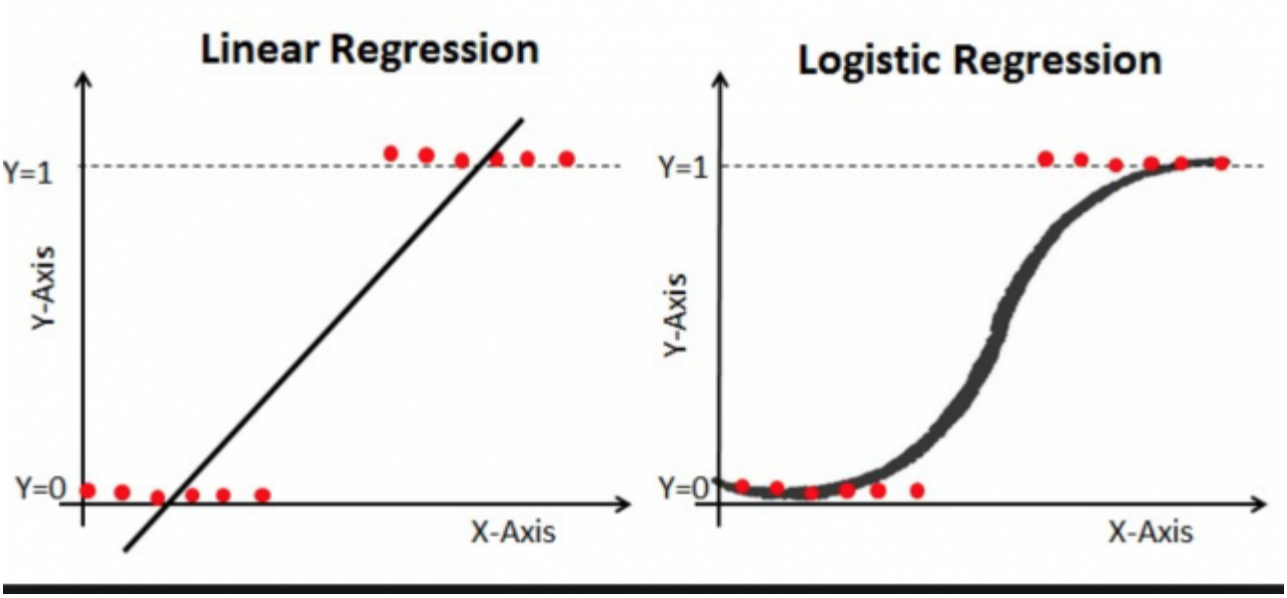
KNN is a classification model that classifies data points depending on how similar they are to each other. It makes a "informed judgment" on what an unclassified point should be classed as based on test data.
KNN is a lazy learning method that is both non-parametric and non-parametric. What exactly do these two phrases imply?

It is non-parametric since it does not make any assumptions. Rather than presuming that the model's structure is typical, it is built totally from the data provided.

The term "lazy learning" refers to an algorithm that makes no generalizations. This means that applying this strategy requires very little training. As a result, when employing KNN, all of the training data is also used in testing.

3.4 MATHEMATICAL MODEL PROPOSED

- **Support Vector Machine**



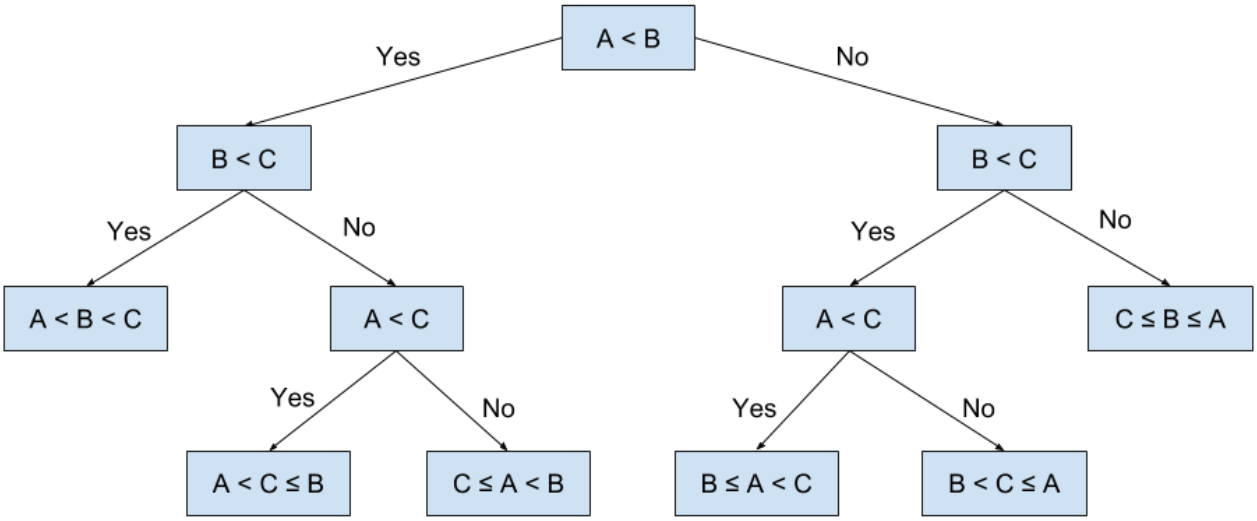
- **Naive-Bayes Algorithm**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

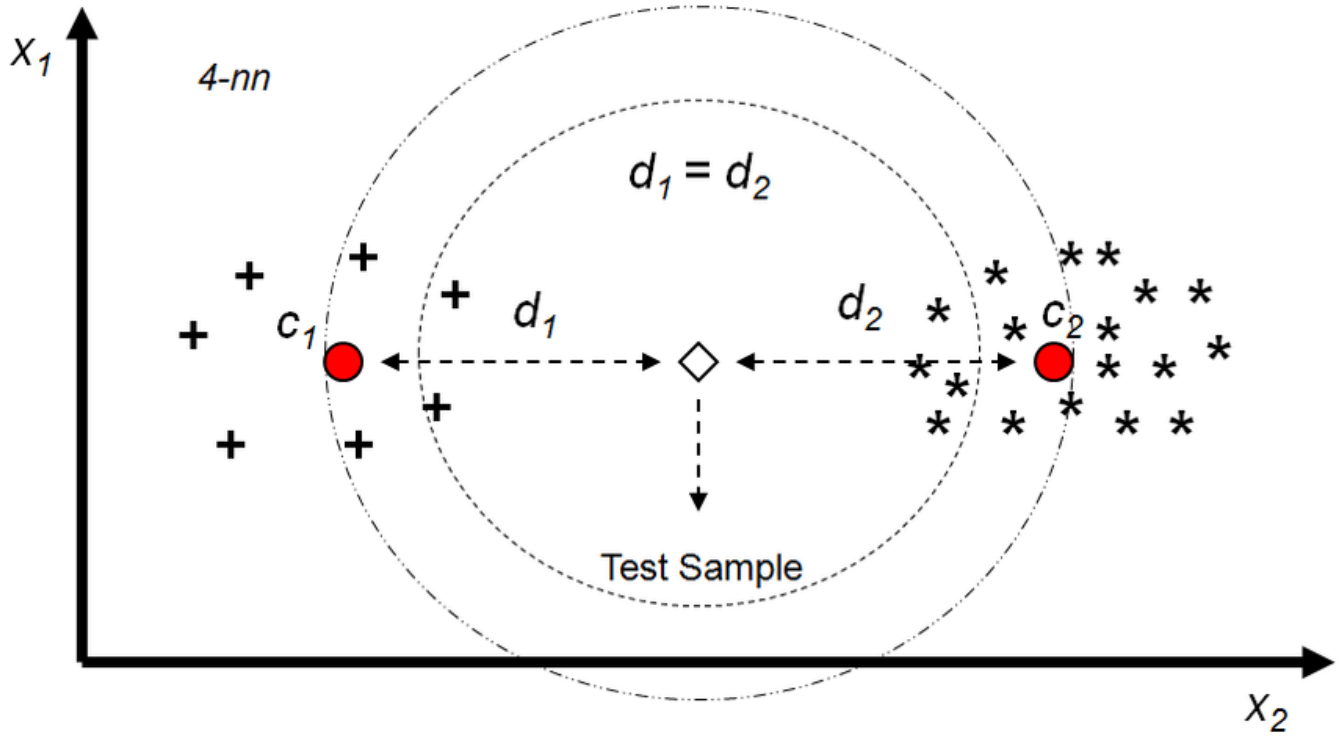
Posterior — Likelihood — Prior — Normalizing constant

$$P(B) = \sum_{Y} P(B|A)P(A)$$

- **Decision Tree**



- **KNN**



4.RESULTS AND DISCUSSION

The attack classes are separated into four groups, with one normal category exhibiting no signs of attack. These four groups, namely

1. **DOS**
2. **Probe**
3. **R2L**
4. **U2R**

These classes have varying assault frequencies, and as seen in the diagram, the DOS attack is the most common among the other forms of attacks.

Model	**Accuracy**
Naïve Bayes	97.37 %
Decision Tree	99.97 %
KNN	99.65%
Logistic Regression	98.08 %

Type of Scores/Models	**Precision**	**Recall score**	**F1-score**
Naive Bayes	0.97	0.97	0.97
Decision Tree	1.0	1.0	1.0
KNN	1.0	1.0	1.0
Logistic Regression	0.98	0.98	0.98

Fig. 5 Evaluation metrics of classifiers

The above table shows that the Decision Tree Classifier and KNN have the highest scores as compared to the other models.

5. CONCLUSION

The second line of defense is Intrusion Detection Systems (IDS). It identifies the existence of assaults in traffic that enters the firewall through the loophole formed in it. An Intrusion Detection System (IDS) continuously observes actions in a given environment and determines whether they are part of a hostile attack or a legitimate use of the environment. The intrusion detection and intrusion avoidance domains are incredibly powerful, with new capabilities, models, and discoveries being made all the time. In addition, a lot of research on information representation strategies for intrusion location data is currently being led. According to the findings of this study, data mining approaches generate fascinating rules that are critical for intrusion detection and prevention in the networking business. We demonstrated how high-performance computing approaches can help with intrusion detection. With the use of a data mining supervised model, this research seeks to solve the problem of intrusion assault detection.

In conclusion, the findings of this research can help to improve networking security.

6. REFERENCES

[1]. R. Heady, G. Luger, A. Maccabe, and B. Mukherjee. A Method To Detect Intrusive Activity in a Networked Environment. In Proceedings of the 14th National Computer Security Conference, pages 362-371, October 1991.

[2]. Biswanath Mukherjee, L Todd Heberlein and Karl N Levitt. Network Intrusion Detection, IEEE Network, May/June 1994, pages 26-41.

[3]. Terresa F Lunt. A survey of intrusion detection techniques. In Computers and Security, 12(1993), pages 405-418.

[4] Vokorokos, L. and A. Baláž. Host-based intrusion detection system. in Intelligent Engineering Systems (INES), 2010 14th International Conference on. 2010. IEEE.

[5] Chauhan, P. and N. Chandra, A Review on Hybrid Intrusion Detection System using Artificial Immune System Approaches. International Journal of Computer Applications, 2013. 68(20).

[6]. Sarmah, A., Intrusion detection systems; definition, need and challenges. 2001, October

[7] Modi, C., et al., A survey of Intrusion detection techniques in cloud. Journal of Network and Computer Applications, 2013. 36(1): p. 42-57.

[8] Latha, S. and S.J. Prakash. A survey on network attacks and Intrusion detection systems. in Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on. 2017. IEEE.

[9] Debar H, Dacier M, Wespi A. A revised taxonomy of intrusion-detection systems.Annales des Telecommunications. 2000;55(7–8): 361-337

[10] Gong Y, Mabu S, Chen C, Wang Y, Hirasawa K. Intrusion detection system combining misuse detection and anomaly detection using genetic network programming. In: ICCAS-SICE. 2009

[11] Posted on Dec 25, 2014 by Robert Moskowitz industry-topics/blog/network intrusion-methods-ofattack#:~:text=Protocol%2DSpecific%20Attacks,%22)%20or%20malformed% 20protocol%20messages.

[12] E. Packel, [internet] 2012 [cited 2013 July 10]. Available from: http://www.databreachlegalwatch.com/2012/05/cyber-warfare-and-collateral-damage-flame-malware-heats-up-data-security-threat/

[13] ESET. 2012 [cited 2013 July 10]. Available from: http://go.eset.com/us/threat-center/

[14] E. Messmer, Unique malware samples broke the 75 million mark in 2011 [internet]. 2012 [cited 2013 July 10].Available from: http://www.networkworld.com/news/2012/022112-mcafee-malware-report-256316.html

[15] Computer Economics [internet]. 2007 [cited 2013 July 10]. Available from: http://www.computereconomics.com/article.cfm?id=1225

[16] Vilela, Douglas W. F. L.; Lotufo, Anna Diva P.; Santos, Carlos R. (2018). Fuzzy ARTMAP Neural Network IDS Evaluation applied for real IEEE 802.11w data base. 2018 International Joint Conference on Neural Networks (IJCNN). IEEE. doi:10.1109/ijcnn.2018.8489217. ISBN 9781509060146.

[17] Dias, L. P.; Cerqueira, J. J. F.; Assis, K. D. R.; Almeida, R. C. (2017). Using artificial neural network in intrusion detection systems to computer networks. 2017 9th Computer Science and Electronic Engineering (CEEC). IEEE. doi:10.1109/ceec.2017.8101615. ISBN 9781538630075.

[18] Inc, IDG Network World (2003-09-15). Network World. IDG Network World Inc.

[19] Brandon Lokesak (December 4, 2008). "A Comparison Between Signature Based and Anomaly Based Intrusion Detection Systems" (PPT). www.iup.edu.