International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319-6734, ISSN (Print): 2319-6726
www.ijesi.org ||Volume 9 Issue 5 Series II || May 2020 || PP 44-52

Diagnosis and Classification of Malware Using Naive Bayes Algorithm

Dhaval Patel
Assistant Professor Department of Computer Engineering SPCE, Bakrol, India

Abstract: *This work falls in the area of collaborative malware detection systems which depend on different antivirus software for malware detection. In this paper, I propose a decision model based on Naïve Bayes Algorithm, where malware decisions are made based on probability of malware and goodware from participating of antiviruses. I evaluate our proposed work using training data sets and demonstrate malware detection techniques can improve the malware detection accuracy.*
Keywords: *Data Mining, Malware, Goodware, Classification, Probability, Naive Bayes, Eclipse, Java.*

Date of Submission: 09-05-2020 Date of Acceptance: 22-05-2020

I. INTRODUCTION

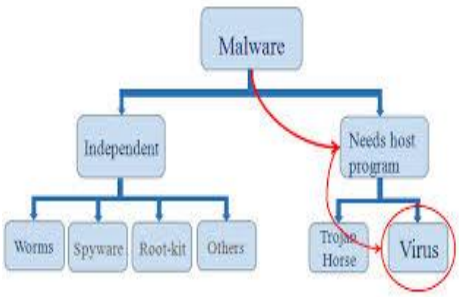

Figure 1 Malware Classification [12]

Data Mining is a technology which helps any organization to process data through algorithms to collect meaningful patterns from large databases. It provides a means of extracting previously unknown, predictive information from the base of accessible data in data warehouses. Data mining tools use sophisticated, automated algorithms to discover hidden patterns, correlations, and relationships among organizational data. These tools are used to predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. Malware is a set of instruction or a computer program which is develop to damage computer system or to harm computer system. The Malware are created purposefully by Malware author. The Malware are also known as a computer virus. Malware is a shorter form of **Mal**icious Soft**ware.** Malware include viruses, adwares, spywares, worms, Trojans, backdoors, bots, rootkits etc. Malware is classified in basically two categories like a) Need Host Program and b) Independent Program. In Need Host Program it includes Trapdoor, Logic Booms, Trojan, and Trojan. When in Independent Program it includes Bacteria and worms. There are different Data Mining techniques used for Malware Detection. The techniques are like Classification, Clustering, and Association Rule etc. there are different methods available for Malware detection like Signature Based Malware Detection, Heuristic Based Malware Detection, Behaviour Based Malware Detection etc.

II. LITERATURE SURVEY

During the literature review analysis was done on the various used approaches, frameworks, algorithms. This related work proposed idea about different technique used for malware detection and its parameter like efficiency, accuracy, time taken to detect suspicious file etc.

TITLE	APPROACH	SUMMARY
proposal of a method detecting malicious processes [1] IEEE, (2014).	Static and Dynamic analysed method	Main focus to check process generate suspicious communication is malware or not. Still to improve effectiveness with implement prototype of proposed system for

		better effectiveness of wild malware.
Malware Detection by Text and Data Mining[3] IEEE,(2013)	Text Mining with feature selection	Detect malware base don API sequence call. Proposed novel ways to detect malware using text and data mining. using feature selection followed by SVM yielded **100% sensitivity**
Detection Of Malicious Transaction In Database Using Log Mining Approach[4] IEEE,(2014)	Log Mining Approach.	It can achieve desired true and false positive rates when confidence and support are set appropriately. It maintains data dependency rules set and optimize the performance of malware detection.
Adaptive Worm Detection Model Based on Multi classifiers[5] IEEE,(2013)	Anomaly Behaviour Approach and multi classifier	Proposed WDMAC model for worm detection to detect known/unknown worm and also to achieve higher accuracy and detection rate.
Malware Detection Based On Objective-Oriented Association Mining[6] IEEE,(2013)	Object Oriented Aassociate Mining Approach	Proposed an API based Data Mining method for detect unseen malware. Frequent item set is evaluated by its support and its classification capability. This method proves that proposed method is effective and able to detect unseen malware.
Malware Detection by Data Mining Techniques Based on Positionally Dependent Features[7] IEEE,(2010)	Heuristic Malware Detection Approach	Focusing on processing Static Positionally dependent features which consider the specificities of objects file format of potential malware containers. The paper describes the realization and investigation of the common methodology for design of Data Mining-based malware detectors' using Positionally dependent static information**.**
An Efficient and Robust Decision Model for Collaborative Malware Detection[8] (Base Paper) IEEE,(2014)	RevMatch Model	Proposed Novel decision model ,where collaborative malware decision made based on labeled malware detection history from participating Anti-viruses

Table 1 Literature Survey

III. RELATED WORK

In my research work, malware can be detected with the help of many function and technique. Here I work with the method of RevMatch Model and Naïve Bayesian Classification algorithm for malware detection. I create training data set for our implementation and on the base of that I detect suspicious file and classify that whether it is "Malware" or "Goodware". In training set I set different antivirus manually. The different anti-viruses give feedback of scanned file and decided that the scanned file is suspicious file or not. On the based of that result I classified the scanned file into the category of "Malware" and "Goodware". There are possibilities that some times some of anti-viruses can't give feedback, some times antivirus gives equal possibilities of Malware and Goodware and in that case system ignores that feedback. As our proposed work I solve equal possibilities problem using Naïve Bayesian Classification techniques.

3.1 Expected outcomes of related work

The expected outcome of the proposed work which is based on Naïve Bayesian Classification for malware and goodware detection and classified like, the malware is detected from different anti-viruses when alarm rise =1 with high TP rate and low FP means it is a malware detected and when alarm rise=0 means no malware detect or it is a goodware with TN rate.

3.2 Evaluation Parameter

I proposed criteria as experimental parameters for malware detection which are:

- **FP:** False Positive is a metrics used to measure quality of malware.
- **TP:** True Positive is a metrics used to measure quality malware.
- **TN:** True Negative is a metrics used to measure quality of goodware.
- **Quality Score:** Quality score is a quality of malware and goodware detection for each and every antivirus.
- $\mathbf{P_M}$**:** Probability of Malware
- $\mathbf{P_G}$**:** Probability of Goodware

IV. PROPOSED ALGORITHM

Step 1: study area of collaborative malware detection system
Step 2: study of compare different decision models to decide whether given file is infected or not.
Step 3: comparison of all decision models to check accurate model among all model.
Step 4: collect data set for malware analysis system from different antivirus vendors.
Step 5: feedback from different antivirus filter those feedback value to get accurate decision.
Step 6: based on Naïve Bayesian Classification modified by filtering values and get accurate decision which is helpful to take decision.
Step 7: base on several evaluation metrics generated by Naïve Bayesian, I measure probability and feedback try to get better result.

V. SYSTEM FLOWCHART

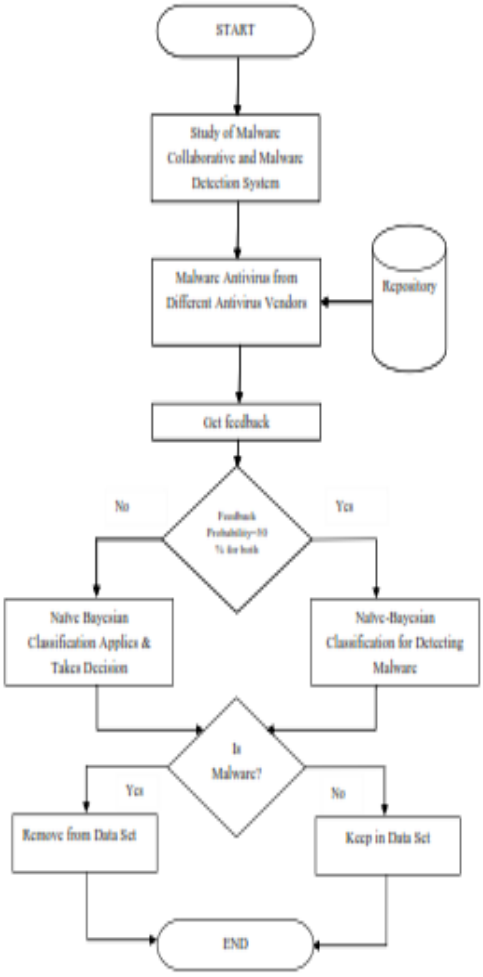

Figure 2 *System Flow*

VI. DESIGN CONSTRAINTS

1. Minimum Hardware Specification

Laptop, computers with T6570 @ 2.10GHz Intel(R) Pentium(R) Core 2 Duo CPU processor and 2 GB RAM having Windows XP (32 bit) Operating system, x64 – based processor is required for the working of above proposed work.

2. Minimum Software Specification

All Algorithms are implemented in eclipse, so for our proposed work I installed implementation tool eclipse.

VII. IMPLEMENTATION ENVIRONMENT

Eclipse:

- **Eclipse** is an integrated development environment (IDE). It consists of base workspace and an extensible plug-in system for customizing the environment in java programming. The Eclipse SDK involves the

Eclipse Java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files.
The main advantages of using eclipse for implementation include:
(i) **Code Completion:** instead of going through documentation I should able to tab our way through methods and save lots of writing.
(ii) **Syntax Checking:** help out to write correct code while typing.

- Our proposed work for malware detection techniques it is carried out with ECLIPSE tool for implementation and generates result.

Language Specification

- For our proposed work I used java as a programming language. I implement Naïve Bayesian algorithm in java programming for calculate probability of malware and goodware.

Data Set

- For our experiment I create training dataset for implementation.
- I create no. of test case to check whether the file is malware or goodware.
- The test case created in .arff format.

Input: Number of files uploads to check file is malware or goodware.
Output: give result in form of whether file is malware or file is goodware.

VIII. EXPERIMENT SETUP

- Tool used: Our experiment performs in Eclipse tool.
- Technology: The training data set has been import to the eclipse with java programming.
- To calculate probability of uploaded file to check whether it is malware or goodware I used Naïve Bayes Classification algorithm.
- After choosing Naïve Bayes classifier it gives result whether uploaded file is malware or good ware on the base of probability.
- If probability of malware (P_M) file is greater than 50% then it consider that file is a malware and probability of goodware (P_G) file is greater than 50% then it consider that file is goodware.

IX. RESULT ANALYSIS

9.1 Quality Score of Antivirus

The following different result shows that file which has been performed on eclipse is malware and good ware.

Antivirus	Quality Score
AV1	0.82
AV2	0.81
AV3	0.805
AV4	0.825
AV5	0.814
AV6	0.818
AV7	0.811
AV8	0.822
AV9	0.813
AV10	0.827
AV11	0.837
AV12	0.824
AV13	0.885
AV14	0.855
AV15	0.84
AV16	0.844
AV17	0.829
AV18	0.807
AV19	0.801
AV20	0.789
AV21	0.704
AV22	0.701
AV23	0.699
AV24	0.68
AV25	0.67
AV26	0.553
AV27	0.543
AV28	0.52
AV29	0.5
AV30	0.49

AV31	0.39
AV32	0.37
AV33	0.33
AV34	0.44
AV35	0.29
AV36	0.24
AV37	0.18
AV38	0.12
AV39	0.08
AV40	0.05

Table 2 *Quality Score of Antivirus*

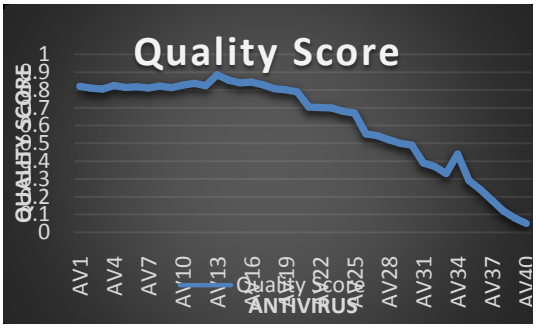

Figure 3 *Quality Score of Antivirus*

9.2 File is Malware

```
=== Classifier model (full training set) ===

ZeroR predicts class value: Malware

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          27               69.2308 %
Incorrectly Classified Instances        12               30.7692 %
Kappa statistic                          0
Mean absolute error                      0.4296
Root mean squared error                  0.4616
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               39

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0          0         0         0          0.5      GoodWare
                 1         1          0.692     1         0.818      0.5      Malware
Weighted Avg.    0.692     0.692      0.479     0.692     0.566      0.5

=== Confusion Matrix ===

  a  b   <-- classified as
  0 12 |  a = GoodWare
  0 27 |  b = Malware
```

Figure 4 *Malware Detected*

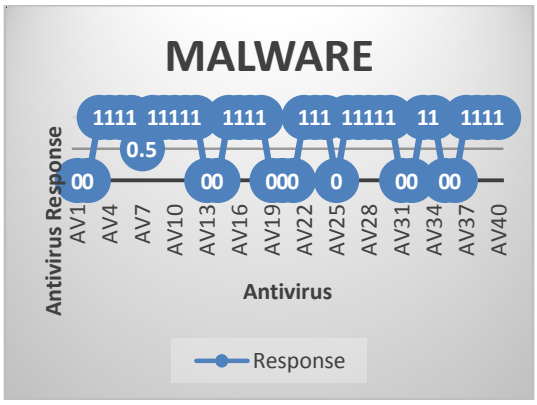

Figure 5 *Malware Detection Graph*

9.3 File is Goodware

```
=== Classifier model (full training set) ===

ZeroR predicts class value: GoodWare

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          27               69.2308 %
Incorrectly Classified Instances        12               30.7692 %
Kappa statistic                          0
Mean absolute error                      0.4296
Root mean squared error                  0.4616
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               39

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.692     1         0.818      0.5      GoodWare
                 0         0          0         0         0          0.5      Malware
Weighted Avg.    0.692     0.692      0.479     0.692     0.566      0.5

=== Confusion Matrix ===

  a  b   <-- classified as
 27  0 |  a = GoodWare
 12  0 |  b = Malware
```

Figure 6 *File is Goodware*

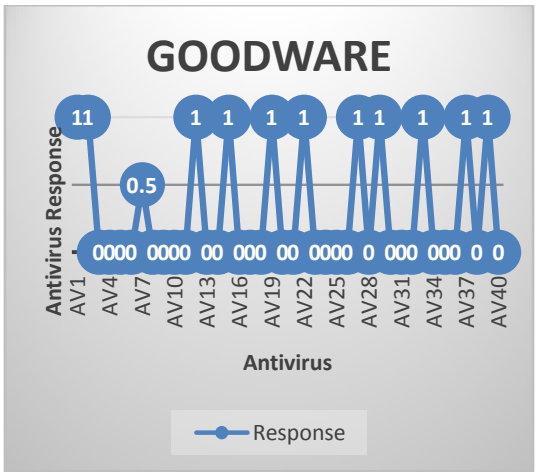

Figure 7 *Goodware Representation graph*

9.4 Equal (50%-50%) probability as a proposed work

```
                Class
Attribute      GoodWare  Malware
                  (0.5)    (0.5)
================================


Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          19               50      %
Incorrectly Classified Instances        19               50      %
Kappa statistic                          0
Mean absolute error                      0.5
Root mean squared error                  0.5
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               38

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.5       1         0.667      0.5      GoodWare
                 0         0          0         0         0          0.5      Malware
Weighted Avg.    0.5       0.5        0.25      0.5       0.333      0.5

=== Confusion Matrix ===

  a  b   <-- classified as
 19  0 |  a = GoodWare
 19  0 |  b = Malware
```

Figure 8 *Equal Detection of Malware and Goodware*

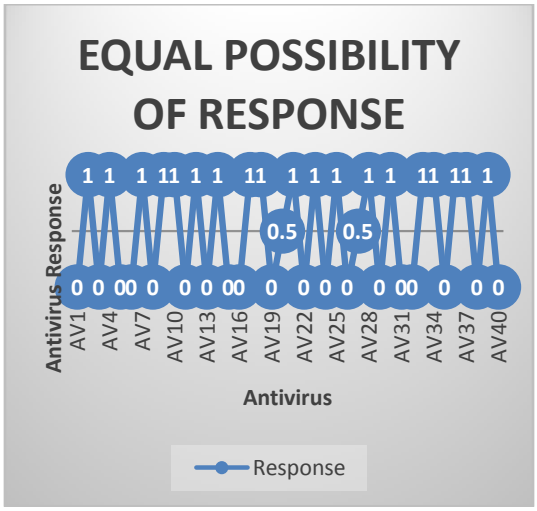

Figure 9 *Equal Probability graph Representation of Malware and Goodware*

9.5 Based on Equal Probability Malware Detected

```
                Class
Attribute       GoodWare  Malware
                  (0.38)   (0.63)
===================================


Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          19               63.3333 %
Incorrectly Classified Instances        11               36.6667 %
Kappa statistic                          0
Mean absolute error                      0.4667
Root mean squared error                  0.482
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               30

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0          0         0         0          0.5      GoodWare
                 1         1          0.633     1         0.776      0.5      Malware
Weighted Avg.    0.633     0.633      0.401     0.633     0.491      0.5

=== Confusion Matrix ===

  a  b   <-- classified as
  0 11 |  a = GoodWare
  0 19 |  b = Malware
```

Figure 10 *based on Equal Probability Malware Detected*

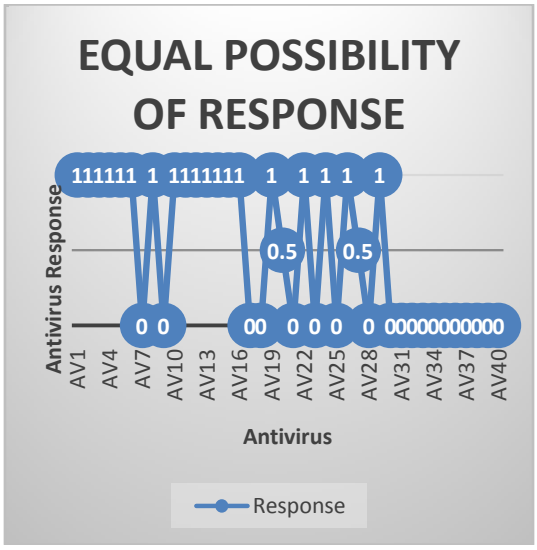

Figure 11 *based on Equal Probability Malware graph Representation*

9.6 Based on Equal Probability Goodware Detected

```
                Class
Attribute       GoodWare  Malware
                  (0.56)   (0.44)
================================


Time taken to build model: 0.01 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          17               56.6667 %
Incorrectly Classified Instances        13               43.3333 %
Kappa statistic                          0
Mean absolute error                      0.4917
Root mean squared error                  0.4956
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               30

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.567     1         0.723      0.5      GoodWare
                 0         0          0         0         0          0.5      Malware
Weighted Avg.    0.567     0.567      0.321     0.567     0.41       0.5

=== Confusion Matrix ===

  a  b   <-- classified as
 17  0 |  a = GoodWare
 13  0 |  b = Malware
```

Figure 12 *based on Equal Probability Goodware Detected*

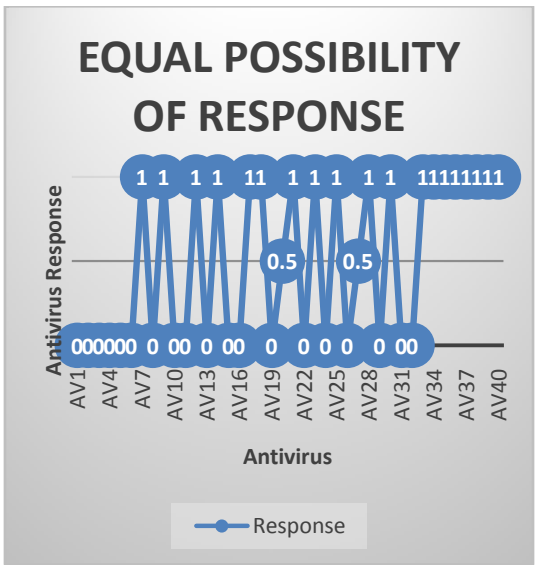

Figure 13 *based on Equal Probability Goodware graph Representation*

X. CONCLUSION

I have presented an efficient malware and goodware classification algorithm using Naïve Bayesian Classification. It performs probability of number of malware and goodware and on the base of that decided particular file is malware or goodware. It classified accurately the scanned file in the category of Malware and Goodware. I solved equal probability of Malware and Goodware problem using Naïve Bayesian Classification algorithm.

REFERENCES

[1]. Takumi Yamamoto, Kiyoto Kawauchi, "Proposal of a method detecting malicious Processes", IEEE, 28th International Conference on Advanced Information Networking and Applications Workshops, pref., 247-8501, 2014.
[2]. Asmitha K A, Vinod P, "A Machine Learning Approach for Linux Malware Detection", IEEE, 2014.
[3]. G. Ganesh Sundarkumar, Vadlamani Ravi, "Malware Detection by Text and Data Mining", IEEE, 2013.
[4]. Ms. Apashabi Chandkhan Pathan, Mrs. Madhuri A. Potey, "DETECTION OF MALICIOUS TRANSACTION IN DATABASED USING LOG MINING APPROACH", IEEE, International Conference on Electronic Systems, Signal Processing and Computing Technologies, 2014.
[5]. Tawfeeq S. Barhoom, Hanaa A. Qeshta, "Adaptive Worm Detection Model Based on Multi classifiers", IEEE, Palestinian International Conference on Information and Communication Technology, 2013.
[6]. XIAO XIAO, DING YUXIN, ZHANG YIBIN, TANG KE, DAI WEI, "MALWAR DETECTION BASEDD ON OBJECTIVE-ORENTED ASSOCIATION MINING", IEEE, Proceedings of the 2013, International Conference on Machine Learning and Cyberetics, 2013.
[7]. Dmitriy Komashinskiy Igor Kotenko, "Malware Detection by Data Mining Techniques Based on Positionally Dependent Features", IEEE, 2010.
[8]. Zahra Bazrafshan, Hashem Hashemi, Seyed Mehdi Hazrati Fard, Ali Hamzeh, "A Survey on Heuristic Malware Detection Techniques", IEEE, 5th Conference on Information and Knowledge Technology, 2013.

[9]. Carol J. Fung, Disney Y. Lam, Raouf Boutaba, David R. Cheriton "RevMatch: An Efficient and Robust Decision Model for Collaborative Malware Detection", IEEE, 2014
[10]. Ashwini Mujumdar, Gayatri Masiwal, Dr. B. B. Meshram, **"**Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches",IJARCET, 2013.
[11]. Carol J. Fung, Quanyan Ahu, Raouf Boutaba, Tamer Basar, "Bayesian Decision Aggrigation in Collaborative Intrusion Detection Networks", IEEE, 2010

Web links

[12]. Fayyad, Piatetsky-Shapiro, Smyth, "From Data Mining to Knowledge Discovery: An Overview", http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html
[13]. Digital lesions, http://www.techstin.com/category/basic-lessons/page/2/
[14]. Malware Repartition, http:// www.securelist.com

Dhaval Patel, et. al. "Diagnosis and Classification of Malware Using Naive Bayes Algorithm." *International Journal of Engineering Science Invention (IJESI),* Vol. 09(05), 2020, PP 44-52.