International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-8, July 2022

Ensemble Filter Technique for Detection and Classification of Attacks in Cloud Computing

Darshan Thakur, Tanuja Pattanshetti

Abstract*: In all technologies, including traditional computing and cloud computing, security has always been the primary concern. In recent years, cloud computing has become widely accepted on a global scale. Cyber attacks aimed at it have increased along with its widespread acceptance. Although ample research is done in the security domain and cloud computing is based on rigid security fundamentals, the advancing network security attacks create the need for an advanced security mechanism. Also, the multiclass classification strategy has received very little attention, and classification accuracy can yet be improved. Hence, this work proposes an Ensemble Filter-based Intrusion Detection System (EFIDS) to address the limitations of previous research work. It not only identifies malicious traffic but also categorizes the attempted attacks (multiclass classification). The famous intrusion detection benchmark dataset, NSL KDD, is used to evaluate the model. Using the model, it was possible to enhance the classification accuracy of both binary and multiclass approaches up to 99.85 percent and 99.63 percent, respectively. Additionally, both forms of classification have shown a 65–70% improvement in training time.

Keywords: Cloud Computing, Security, Feature Filter, Classification

I. INTRODUCTION

Ever since Information Technology has been evolving, each new invention has faced security issues from day one. Cloud Computing is one of the technologies that acquired a place in mainstream technological demands in recent years. Consequently, the cloud has become one of the most common targets of interest for cyber attackers. As cloud computing is a technology wherein the resources are hosted on a network, it involves the flow of packets. These packets are the medium that attackers use to exploit the cloud. Thus, all the requests (packets) that the network receives are not always legitimate or benign. Various types of attacks in cloud computing are:

- **Neptune**: Exploits the SYN Packet vulnerability of TCP protocol Three-way handshake.
- **Ipsweep**: The reply of ICMP echo requests reveal the target's IP address. IP sweep aims to determine which range of IP addresses map to live hosts.
- **Portsweep**: ICMP echo requests are sent to identify the open ports through the responses.
- **Smurf**: A smurf attack is a form of a DDoS attack that causes packet flood on the victim by exploiting/abusing ICMP protocol.
- **Teardrop**: A teardrop attack is a denial-of-service (DoS) attack that involves sending fragmented packets to a target machine.

Providing protection to the system attack-wise is not feasible, as it would lead to the increased complexity of security mechanisms creating other issues in terms of network latency and various other performance aspects. Various security solutions are available, among which the most comprehensive solution is – the "Intrusion Detection System (IDS)."

A. Intrusion Detection System

The IDS is an application or device that works as a network monitoring system to detect any violation of network rules or security. It keeps a watch on every packet or request that enters the network and detects whether it is safe for the network or not. If that packet is not found to be harmless for the system, then it is reported to the security administrator to make the decisions over it. Unlike a firewall, it does not block the incoming and outgoing connections, but only checks if the system is being compromised or not by monitoring the network traffic. However, few IDS that can respond to specific scenarios do exist. Evidently, Intrusion Detection System is an important security mechanism that not only cloud computing but any technology involving networking would require. But, over the past decade cyber-attacks have become smart, fast, and complex. To keep up with these advancing attacks IDS also need to become efficient and faster. "Fig. 1" Many researchers have provided the solutions that integrate Machine Learning into Intrusion Detection systems (shown in Fig. 1). To help the research various benchmark datasets like KDD'99, NSL KDD, etc. are available.

Manuscript received on 29 June 2022 | Revised Manuscript received on 04 July 2022 | Manuscript Accepted on 15 July 2022 | Manuscript published on 30 July 2022.
***** Correspondence Author
Darshan Thakur*, Department of Computer Engineering, College of Engineering, Pune (Maharashtra), India. Email: thakurds20.comp@coep.ac.in
Dr. Tanuja Pattanshetti, Department of Computer Engineering, College of Engineering, Pune (Maharashtra), India. Email: trp.comp@coep.ac.in

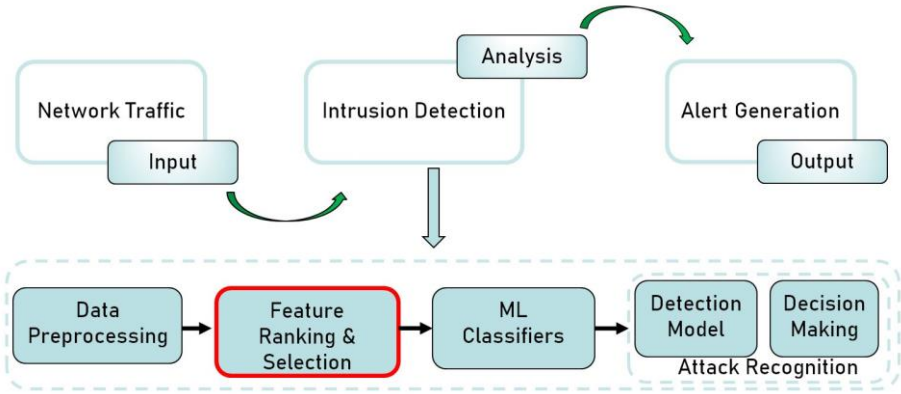

Fig. 1. Integration of Machine Learning in IDS

These datasets are used to test the machine learning models in terms of accuracy, prediction time, false positives, and many more criteria. A Benchmark dataset is a collection of data entries that represent various scenarios of a real-time environment. It helps researchers to test their models without the need for exact resources and environment wherein the model is meant to be deployed.

B. Feature Selection

In this era of Big Data, benchmark datasets not only contain a large number of instances but also numerous features. Each feature in the dataset has certain amount of correlation and contribution to the class label or dependent feature for prediction. Some features are not or less required for prediction, due to various reasons like redundancy, low importance. Process of selecting important and relevant set of features from the dataset is called as Feature Selection. In this process, certain statistical methods or machine learning models are used to identify the importance of each feature of dataset in prediction of class label. Once the importance level of each feature is known a feature subset can be generated that would be better than all-feature dataset.

C. Importance of Feature Selection

Feature selection has many advantages that make it an important step in machine learning. It removes the irrelevant and noisy features by retaining those with the least redundancy and most relevance to the target feature. Moreover, it helps improve the performance of machine learning algorithms, avoid over-fitting and reduce their training, testing, and computational time. Another important benefit of feature selection is that it helps in the reduction of the dimensionality which makes the further process simpler for researchers. Two prominent methods of feature selection are discussed below:

1) Filter Method

In Filter-based feature selection methods, a subset of features is obtained based on their relationship with the class label. Identification of subset simply involves ranking or ordering the features from best to worst. The ranking is done based on the various intrinsic properties of the data, such as consistency, variance, distance, correlation, information, etc. This process is completely independent of the recursion in learning algorithm that is going to be used. Thus, filter methods are less complex and produce results faster than other methods, making them the most widely accepted and used feature selection methods. Since it requires the target variable, it is a type of supervised feature selection method. "Fig. 2." describes the filter technique process.

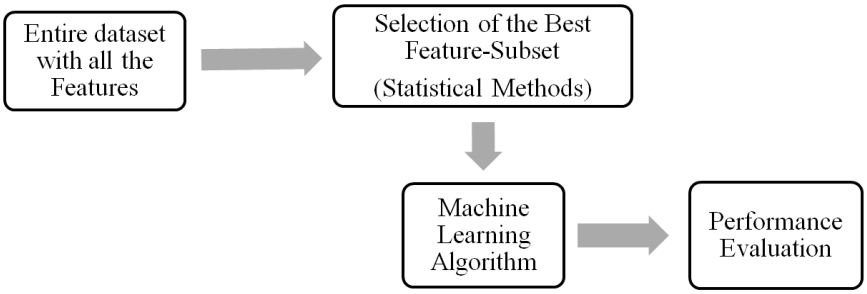

Fig. 2. Filter Technique

Few Filter methods are:

- **Information gain**: Often used in training decision trees. Help in reducing the entropy or surprise.
- **Chi-square test**: Obtained by diving the square difference between actual and expected data values by the expected data values.
- **Fisher score**: Gradient or the derivative of the log likelihood function is used to allot the score to each feature.

Retrieval Number: 100.1/ijitee.H91800711822
DOI: 10.35940/ijitee.H9180.0711822
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

• **Correlation coefficient**: Correlation coefficients are generally used to measure the strength of a relationship between two variables.
• **Variance threshold**: Removes the low variance features based on a specific threshold value.

2) Wrapper Method

In this feature selection method, the feature searching strategy is wrapped with an ML algorithm. The role of the machine learning algorithm here is the evaluation of each feature subset to finally find the best one. It is like a black box evaluator for search strategy. Unlike filter methods, they do not rank features but find the most well-performing feature set. These methods are easy to implement but when the count of features in the dataset is high, it consumes an exceptional amount of resources and computational time. Hence it is preferred only when the availability of resources is not an issue. It is also a type of supervised feature selection method. "Fig. 3." describes the general wrapper technique process.

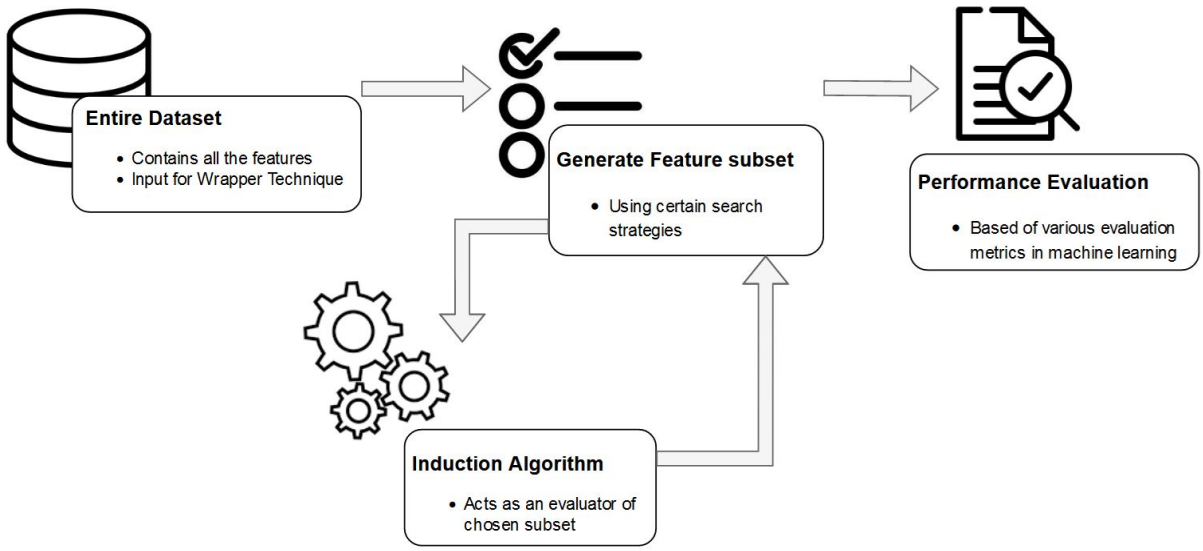

Fig. 3. Wrapper Technique

Few wrapper methods are:

• **Recursive feature elimination**: It recursively eliminates the irrelevant feature until the desired number of features are obtained.
• **Sequential feature selection algorithms**: The algorithm selects multiple features from the full set of features and evaluates them for model and iterates number between the different sets.
• **Genetic algorithms**: Genetic algorithms use an approach to identify an optimal feature set based on evolution.

II. MATERIALS AND METHODS

A large number of Intrusion Detection models have been proposed for cloud computing, Anupama Mishra [1] proposed a ML based approach using classification techniques for identification of DDoS attacks in the cloud computing environment.

Three classifier algorithms namely, Random Forest (RF), Naive Bayes (NB), and K Nearest Neighbor (KNN) are used. V. Bol'on-Canedo [2] tested various combinations of discretizers, filters, and classifiers for multiclass classification. Shih-Wei Lin [3] used simulated annealing and SVM for feature selection and combined the decision tree to build the decision rules. The system provides high prediction accuracy and also decision rules generated from it, help to detect a few unseen attacks. Wrapper-based feature selection is a widely accepted technique to reduce the number of features, Md. Monirul Kabir [4] proposed a wrapper-based feature selection technique named Constructive Approach for feature selection (CAFS) using a neural network and tested it on eight different benchmark datasets. The model uses the correlation information to select the features and supply them to the Neural Networks thus resulting in a compact neural network with less amount of redundant information. Enrique Romero [5] addressed two decision issues in the wrapper method with sequential backward selection and multi-layer perceptron, the retraining of the network prior to computation of the saliency and secondly the stopping criterion in case of the network training. Wenjuan Wang [6] used the correlation between dependent features and the class label of the dataset for selecting the features that contribute highly to classification in IDS. This filtered dataset was supplied to the Support Vector Machine. Filter-based feature selection requires an optimum number of resources to obtain the feature rankings as compared to the wrapper-based approach. Hee-su Chae [8] proposed a feature selection method that uses Attribute Ratio (AR) i.e. average value and frequency of features. Fengli Zhang [9] discussed the performance of prominent filter methods like chi-square, gain ratio, information gain, ReleifF, and sequential search strategy based on Naive Bayes. An ensemble of these prominent filter methods was done to obtain the feature subset and train on the decision tree by Opeyemi Osanaiye [10] in order to improve the accuracy and training time of the model. Along with the ensemble of filter methods Sivamohan Krishnaveni [11] proposed a system with ensemble at the learning model level as well. It combines the feature subset selection results from three filter methods (IG, GR, and Chi-square) and this subset is further supplied to a majority voting-based classifier which is based on multiple learning models (Logistic regression, Naive Bayes (NB), SVM, and Decision Tree (DT)).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Further, Saikat Das [12] combined seven filter techniques based on majority voting to produce an optimal feature set. Mustapha Belouch [13] proposed a hybrid wrapper filter ensemble feature selection technique that combines the benefits of both. Improvement in overall efficacy is observed from the results when compared with individual filter and wrapper approaches. Evidently, the topic of feature selection has been the subject of substantial investigation. The filter method is the most popular technique among feature selection methods because of its portability, ease of use, and low computing complexity. In most cases, a subset with about one-fourth of the features from the original dataset is extracted, and regardless of the learning model used, the findings show better accuracy. However, choice of the ideal prediction model yields much greater outcomes. Numerous methods based on combining prediction models and feature selection methods have been proposed, and the results are fairly accurate.

III. THEORY

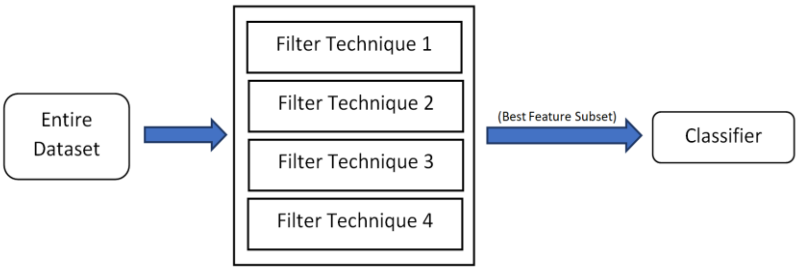

Fig. 4. Basic structure of EFIDS

An ensemble of various filtering algorithms is used in the proposed system, Ensemble Filter-based Intrusion Detection System (EFIDS), which uses the complete dataset as input and outputs the best feature subset. "Fig. 4." shows the basic structure of the system. Following is a detailed discussion of the most well-known and widely used filtering techniques:

A. Information Entropy

Entropy measures the disorder of the system. A system with high entropy would be unpredictable and more disordered and would be less surprising. On the other hand, a system with less entropy would be highly predictable and would have less disorder. In the context of Data Science, Information Entropy measures how unpredictable a data distribution is. Mathematically, Entropy is defined as:

$$H(A) = -\sum_{i} x_i \log_2 (x_i)$$

where, *i* is the number of different values that A can take and *A* is the feature of which entropy is calculated.

B. Information Gain (IG)

Information gain is the amount of entropy (disorder) that is removed by knowing an input feature beforehand. Mathematically, Information gain is defined as,

$$IG(B/A) = H(B) - H(B/A)$$

Here the formula indicates the information gain of instance B given that an input A is known. The more the Information gain, the more entropy is removed, and the more information does the variable *A* carries about *B*. For feature selection the information gain for each of the features in dataset, IG(X1), IG(X2) and so on are determined. Then the features are ranked in the descending order of their respective information gains. A certain value of the threshold is decided and all the features above the threshold are selected for training the machine learning algorithms. Information Gain method is also used in the decision tree algorithm to decide the splitting criteria.

C. Gain Ratio

Gain ratio (GR) is an optimized version of the information gain that helps in the reduction of its biased nature. In the Gain ratio, the count and length of the branches are taken into consideration while selecting a feature. It optimizes the IG by considering the inherent details of a split. Here the inherent detail is the information entropy resulting when instances are distributed across branches (i.e. amount of information required to identify the branch an instance belongs to). When inherent detail increases, the value of the attribute gets smaller.

D. Gini Decrease

This filter technique makes use of Gini Impurity, also known as Gini index. The probability of a certain feature being classified incorrectly on random selection is used by this filter. It is called pure when every element is linked to one class. As seen in the case of entropy, Gini Impurity also lies in the range from 0 to 1. If the value of impurity is 0, it indicates that the classification is pure which means all the elements lead to a certain class or only a single class exists.

While if the value is 1, it means that the distribution of instances across different classes is random in nature. If the impurity value is 0.5 it indicates that instances are distributed equally across few classes. The features with lowest value of impurity are given preference when constructing the decision tree. Mathematically, Gini Impurity is one minus summation of squared probabilities of every class i.e. the expression is:

$$Gini\ index = 1 - \sum_{i=1}^{n} (P_i)^2$$

where, P_i is the probability of an element being classified for a distinct class.

E. Chi-Square (X^2)

In statistics chi-square test is used to identify the dependency of two events. For any two features given their values, expected count *E* and observed count *O* can be obtained. The deviation between observed count and expected count is given by Chi-Square.

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where,

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

O = value(s) observed, c = degree of freedom, E = value(s) expected

When the expected count and observed count are close, it indicates the considered two variables are independent. Thus, hypothesis of independence is set wrong when the value of chi-square is high. Consequently, when the value of chi-square between certain feature and class label is high, the feature is highly dependent on the outcome and hence can be given higher rank.

F. ReliefF

The discovery of feature value differences between nearest neighbor instance pairs forms the basis of relief feature scoring. A difference in values of feature in pair of neighboring instances, with the identical class values decreases the feature score. While, if that difference is noted in pair of neighboring instances with distinct class values, then there is increase in feature score. The first case is said to be a hit while the latter a miss. The technique is applicable when the class label is binary in nature. For multiclass type of label, a variant of Relief called ReliefF is used. The first step in ReliefF is random selection of a feature F_i and then search for N nearest neighbors that belong to identical class C_j followed by that of n nearest miss $F_j(G)$ for every other class. Finally, the weights of all attributes are updated and then the mean of inputs of all misses and all hits.

G. Fast Correlation Based Filter (FCBF)

Dependence between all the feature pairs and class relevance are used by FCBF feature selection that is multivariate. It is based on concept of "Predominant correlation". Features that are slightly correlated to other features and highly to the class label are ranked high. Class relevance and feature dependencies are calculated using Symmetrical uncertainty (SU). When the correlation of a feature with class label is higher than threshold value of SU, that feature is selected or ranked high. Predominant correlation with class label means a feature is correlated to class label maximum and not more than that to any other feature. Further the second-best feature is considered. Hence, given an all set of features the FCBF removes redundant and irrelevant features using sequential search (SS) strategy and backward selection technique (BST). This is continued until the feature list is empty.

IV. BENCHMARK DATASET

The famous network intrusion benchmark dataset, NSL-KDD is used for the training and evaluation of the EFIDS and to compare it with the various existing solutions of same class. An old benchmark dataset, KDD'99 had few drawbacks discussed in [14] by Mahbod Tavallaee and his team. To mitigate few of these issues a data set namely, NSL-KDD was built. It has various advantages over former like, lesser redundancy, reasonable count of entries making it computationally affordable for experimentation and no duplicate records. It represents a diverse network system that includes the network requests based on icmp, tcp and udp protocols. Also it covers 26 types of flags under the flag attribute and there are various types of attacks under the class label; "Fig.5" shows the distribution.

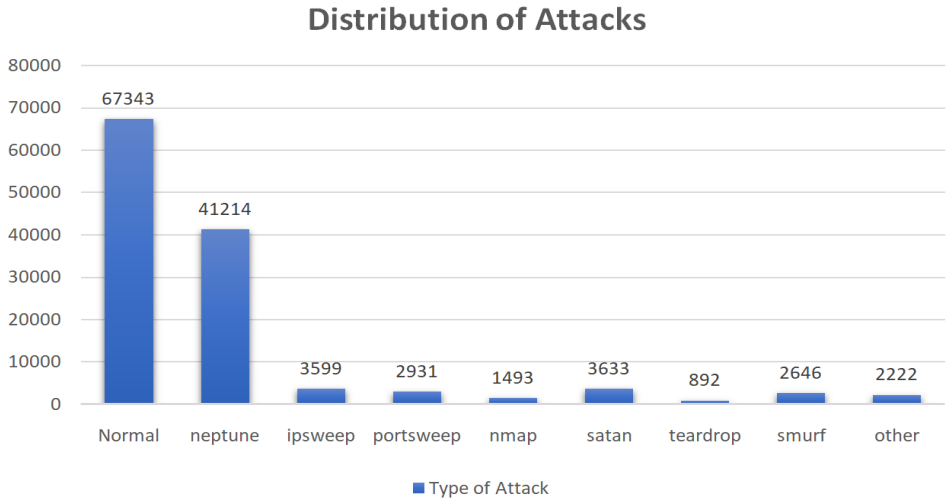

Fig. 5. Distribution of the attacks in NSL-KDD dataset

Table-I: Categorization of attacks in NSL-KDD

Category	Portion	Type of packets
DoS	36.47%	pod, back, teardrop, land, smurf, neptune
R2L	0.79%	phf, ftp_write, warezmaster, guess_password, warezclient, imap, spy, multihop
Probe	9.25%	portsweep, ipsweep, nmap, satan
U2R	0.04%	loadmodule, buffer_overflow, rootkit, perl
Normal	53.45%	harmless/benign traffic

The dataset contains 41 features and 1 class label, which can be divided into 4 types. Features 1-9 are intrinsic types of features which mean that they can be derived from the packet header irrespective of its payload. While the content type of features are 10-22 and they provide the information about the original packet as the packet is transmitted into parts.

This information is used to access the payload. Features 23-31 are Time based features that provides analysis of information like number of connection requests to same host independent of content of the network request. These features are based on two seconds window. Host based features are the last type of features that range from 31-41 and unlike previous type are based on analysis from more than two seconds window.

The dataset has 22 types of attacks in total as class label values and 1 value as normal which refers to the harmless network packets.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The dataset involves Neptune (32.72%), ipsweep (2.86%), portsweep (2.33%), nmap (1.19%), satan (2.88%), teardrop (0.7%), smurf (2.1%), and other 15 types of attacks (1.77%). Rest of the part of dataset has the normal network requests, which cover about 53.45% of all the rows.

These attacks are can be grouped into 4 categories namely: U2R, DoS, Probe, and R2L. "Table-I" shows the distribution of these categories in the NSL-KDD dataset.

A. Denial of Service (DoS)

Availability of a network, especially a cloud network is a very crucial pillar that affects both users and owners of the network. Denial of Service is a category of attack that exploits the availability of the network by making it inaccessible or forcing it to shut down by overloading with a large number of requests, beyond the capacity of the servers hosting that network. The types of DoS attacks differ in the approach used to overload and bring the system down.

B. Remote to Local (User)

In R2L type of attacks, the attacker is not part of the victim network i.e., the network is not directly accessible to him.

Hence, the attacker sends malicious packets remotely to the target network where he cannot be a local user. These malicious packets create the backdoor which provides an illegal network access to exploit it further. The systems which have strong security at the physical level but not at the network level are vulnerable to such attacks.

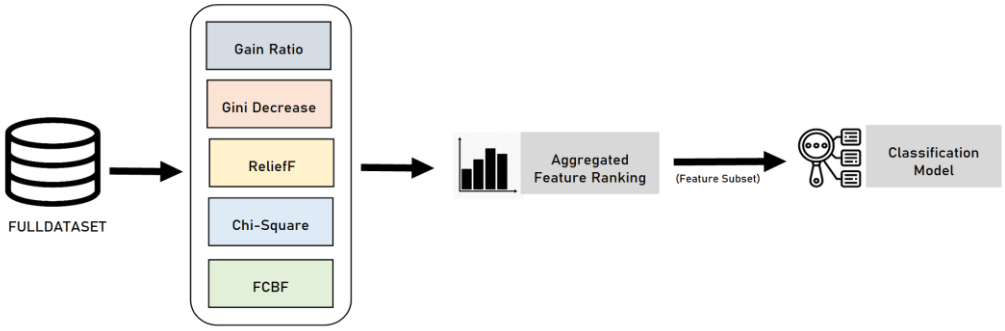

Fig. 6. Structural Block Diagram of EFIDS

C. User to Root (U2R)

These types of attacks, unlike R2L attacks, are the ones in which the attacker has access to the target network. The access may be obtained by actually being a user of the network himself or by gaining the access as some other user. Once the access is gained attacker exploits the system to become the full access user in order to access the system data or make any changes that would harm the network.

D. Probe

In probe attacks, intruder makes use of various tools like nmap, wireshark, and other techniques to monitor the network activity and its state to obtain the crucial information and fingerprints of the network. The intruder then makes use of the obtained information and fingerprints to exploit the network further.

V. CALCULATIONS

EFIDS is an ensemble filter techniques-based intrusion detection system that uses Decision Tree as the classification model. "Fig. 6." shows the structural block diagram of EFIDS. It consists of two phases namely, Feature selection and Classification. The feature selection phase accepts the entire dataset and results into a feature subset which is further given as input to the latter phase.

A. Phase I: Feature selection

In this phase, the full feature train dataset is supplied to various commonly used filter techniques like Gain Ratio, Gini Decrease, ReliefF, Chi-Square, and FCBF.

From each filter, a ranked list of features is obtained. From each of these techniques, Top 60% i.e., a list of 25 top-ranked features is selected. These lists are further merged into a unique single list of features, based on the given formula.

$$u = unique\left(\sum_{1}^{n}\left(f_i(ds)\right)_{top\_60\%}\right)$$

where, n is the number of filter techniques used, ds denote full dataset, f_i denotes ith filter technique For the reduction of the dataset, the Two-Fold approach is applied. So, to reduce the number of features to 21, each feature in this list is assigned a count based on the number of filter techniques it has been selected into the top 60%. Finally, the top 21 features from that list are selected and given as input to the next phase. "Fig. 7." The final selection of 21 attributes from NSL-KDD dataset using EFIDS are shown in the figure 7 along with the count of filter techniques they are top ranked in.

Selected Features	Gini Decrease	ChiSquare	ReliefF	FCBF	GainRatio	Count
dst_host_serror_rate	✓	✓	✓	✓	✓	5
dst_host_same_src_port_rate	✓	✓	✓	✓	×	4
service	✓	✓	✓	✓	✓	5
dst_host_srv_rerror_rate	✓	✓	✓	×	✓	4
flag	✓	✓	✓	✓	✓	5
src_bytes	✓	✓	×	✓	✓	4
dst_host_srv_serror_rate	✓	✓	✓	✓	✓	5
dst_bytes	✓	✓	×	✓	✓	4
logged_in	✓	×	✓	✓	✓	4
dst_host_same_srv_rate	✓	✓	✓	✓	✓	5
count	✓	✓	✓	✓	×	4
dst_host_srv_diff_host_rate	✓	✓	✓	✓	×	4
protocol_type	✓	×	✓	✓	✓	4
serror_rate	✓	✓	✓	✓	✓	5
srv_serror_rate	✓	✓	✓	✓	✓	5
same_srv_rate	✓	✓	✓	✓	✓	5
diff_srv_rate	✓	✓	✓	✓	✓	5
srv_diff_host_rate	✓	✓	✓	✓	×	4
dst_host_srv_count	✓	✓	✓	✓	×	4
dst_host_diff_srv_rate	✓	✓	✓	✓	✓	5
dst_host_rerror_rate	✓	×	✓	✓	✓	4

Fig. 7. Features selected using EFIDS

Retrieval Number: 100.1/ijitee.H91800711822
DOI: 10.35940/ijitee.H9180.0711822
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

B. Phase II: Classification

The second phase accepts the feature subset from phase I as input. On the reduced dataset, the Decision Tree is trained and tested based on a 10-fold cross-validation technique. The two parameters, Train time (amount of time taken to train the model), and classification accuracy, in the case of the reduced dataset are compared with that of the full dataset.

C. Algorithm for EFIDS

Algorithm: Feature selection using the proposed model

***Input**: Full Feature Dataset*

***Output**: 2-fold Feature subset*

1 ***For each selected filter technique f do***
2 $[ranked\_features]_f \leftarrow f(full\_dataset)$
3 $top\_60_f \leftarrow$ *select top 60% features from* $[ranked\_features]_f$
4 ***End***
5 $u \leftarrow$ *select unique features from* $[\sum top\_60_f]$
6 ***For each feature j in u do***
7 j_{count} = *number of* top_60_f *lists* j *occurs in*
8 ***end***
9 $u_{sorted} \leftarrow$ *sort* y *descending on the basis of* j_{count} *of each feature* j
10 ***output*** $\leftarrow$ *select Top 21 features in* u_{sorted}

where,
f: Filter selection techniques in ensemble
j: features
top_60_f: Top 60% features selected from the ranked feature list of filter technique f

VI. RESULTS

The proposed model EFIDS, based on ensemble filter techniques and a classification model: Decision Tree, was tested on the selected benchmark dataset NSL-KDD. The model's results were compared with that obtained using the entire dataset and individual filter techniques. The evaluation metrics used to compare the various cases are classification accuracy and time taken to train the model, i.e., build time. These experiments were conducted for two cases of classification: Binary classification and Multiclass classification. The significant difference between the two is that the outcome possibilities of the model in the case of the former are two in number (whether the network packet is an attack or normal/benign in this case), whereas that in the latter case is more than two (benign or the name of attack which is being attempted through the network packet).

A. Binary Classification:

In the case of binary classification, as seen in the figure, the classification accuracy of EFIDS is high as compared to that of all the individual filter techniques and non-filtered dataset. "Table-II" shows the number of features selected in each case and the respective classification accuracies achieved. EFIDS approach has obtained the highest accuracy of 99.85 percent as compared to other filter techniques and in case of entire dataset.

Table-II: Results of Binary Classification

Filter	Number of Selected Features	CA (%)
Full Dataset	41	99.61
Gain Ratio	21	99.83
Gini Decrease	21	99.71
Chi-Square	21	99.73
Relief F	21	99.35
FCBF	21	99.72
EFIDS	**21**	**99.85**

B. Multiclass Classification

In the case of multiclass classification, EFIDS again showed maximum improvement in accuracy over the entire dataset than any other individual filter technique. "Table-III" shows the number of features selected in each case and the respective classification accuracies achieved.

Table-III: Results of Multiclass Classification

Filter	Number of Selected Features	CA (%)
Full Dataset	41	99
Gain Ratio	21	99.02
Gini Decrease	21	99.6
Chi-Square	21	99.61
ReliefF	21	98.8
FCBF	21	98.86
EFIDS	**21**	**99.63**

The next evaluation metric selected is build time. In binary and multiclass classification, the time taken to build the model was less for EFIDS, as shown in the figure. In the case of multiclass approach, it is as low as 5.28 seconds, which is a 65 percent improvement relative to the time required for the complete dataset (15.18 seconds). In contrast, it takes 3.6 seconds for binary classification, which records a 71 percent improvement from the time needed for the entire dataset (12.65 seconds). "Fig. 8." shows the graphical comparison of the results among EFIDS, various individual filter techniques and full dataset in term of training time.

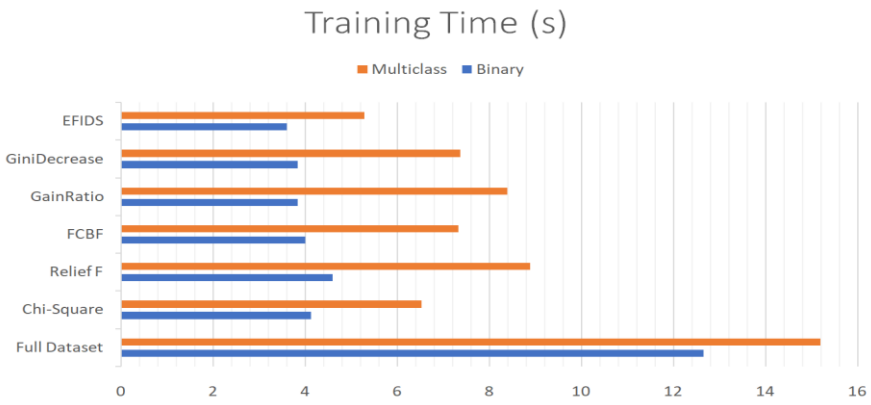

Fig. 8. Training Time comparison of Binary and Multiclass models

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table-IV: Comparison of EFIDS with other approaches

Approach Name	No. of Features selected	Accuracy (%)
Gradual Feature Removal [16]	19	98.62
CFS [17]	NA	99.13
CONS, CFS and INTERACT [18]	7	93.72
CFS and CSE [19]	32	78
Linear Correlation-based [20]	17	99.1
EMFFS [10]	13	99.67
DM based IDS [15]	13	99.75
EFIDS	**21**	**99.85**

"Table-IV" shows the comparison of EFIDS classification accuracy with that obtained from various other approaches proposed in the literature.

Between the two evaluation parameters classification accuracy and training time, the former is given more importance in the experiments and evaluation as the proposed system is meant to be deployed as a security mechanism. Consequently, the accuracy of identifying the malicious requests incoming to the network should be given comparatively more preference than the training time of the model. Hence in comparison with other the approaches, only the classification accuracy is considered. Moreover, considering training time for comparison among the various approaches would not be valid as the experimentation environment for each approach would have been different. The hardware and software configuration used highly influence the time taken to train the model.

VII. CONCLUSION

The proposed model EFIDS makes considerable contributions to the accuracy and training time of the classification model used to detect malicious network traffic, by selecting the important features. The advantage of Filter techniques being the lightest approach among all the types of feature selection methods and the need for lower complexity in Intrusion Detection Systems were brought together provide a simple solution that provides security to the cloud computing network at a lower cost of computation in lesser time. In the future, the aim is to extend the work towards more advanced feature selection techniques and working with the real-time network packets for more reliable training of the system.

REFERENCES

1. Anupama Mishra, B. B. Gupta, Dragan Perakovi´c, Francisco JoseGarciaPenalvo, and Ching-Hsien Hsu. Classification-based ml for detectn. of dist. denialofservice attack in cc. In IEEE ICCE, 2021, pages 1–4, 2021.
2. A.A.Betanzos V.B. Canedo and N.S. Maro˜no. Fs and classification in multi. class data sets: An app. to kdd-cup'99 data set. The Expert Sys. with App., 38(5):5947– 5957, 2011. [CrossRef]
3. Lee Z.J. Lin S.W., Ying K.C. and Lee C.Y. An intelli. algo. with fs and dec. rules applied to anomaly id. Appl. Soft Comp., 12(10):3285–3290, 2012. [CrossRef]
4. Murase K. Mohammad M. Kabir and Mohammad M. Islam. A new wrapper fs approach using nn. The Neurocomputing, 73(16), 2010. Brazilian Symposium on NN, 2008 (SBRN2008).
5. Romero E. and Sopena J. M. Performing fs with multilayer perceptrons. IEEE Trans. on N.N., 19(3), 2008. [CrossRef]
6. Wenjuan Wang, Xuehui D., and Na W. Building a cloud intrusion detectionsystem using an efficient fs methodandsvm. IEEE Access, 7, 2019.
7. L.Yinhui, Silan Z., D. Kuobin, Yan J., Jingbo X., and Xiaochuan A. An efficient ids based on svm and gradually feature removal method. Expert Systems with App., 39(1), 2012.
8. Heesu Chae and Sanghun Choi. Selection for efficient ids using ar.
9. Zhang F. and W. Dan. An effective fs approach for network id. In IEEE Eighth International Conf. on Networking, Arch. and Storage,2013, pages 307–311, 2013.
10. Cai H. Choo KK.R. et al. Osanaiye, O. Ensemble-based multi-filter fs method for distributed denial of service detection in cc. In The J. of Wireless Communication Network, 130 (2016), pages 307–311, 2016.
11. Krishnaveni S., Sridhar S., S. Sivanandam, , and Subramani Prabhakaran. Network id based on ensemble classification and fs method for cc. Concurr. and Computation: Practice and Experience.
12. Saikat Das, Deepak Venugopal, Frederick T. Sheldon, and Shiva S. Empirical eval. of the ensemble framework for fs in distributed denial of service attack. 7th IEEE Intern. Conf. on C. Sec. and CC (CSCloud), 2020.
13. MustaphaBelouch, Salah Elhadaj, and Mohamed Idhammad. A hybrid filter- wrapper fs method for distributed denial of service detection in cc. Intelligent Data Analysis, 22:1209–1226, 12 2018. [CrossRef]
14. T. Mahbod, L. Wei, B. Ebrahim, and A. Ali Ghobrani. A detailed analysis of the kdd-cup'99 dataset. In IEEE Symp. on Comput. Intell. for Sec. & Defense App., year 2009.
15. Ghosh, P., Sinha, S., Sharma, R.R. et al. An efficient Intrusion DS in cloud environment using FS based on Dolphin Mating algo. J ComputVirol Hack Tech (2022). [CrossRef]
16. P. Jian, R Choo Kim Kwang, A. Helen, Bit level n-gram-based forensic authorshipanalysis on social-media: Identifying individuals from the ling. profiles. Elsevier JNetwComput Appicat. (2016 in press)
17. Y. Jaehak, K. Hyunjoong, B. HC, P. DaeHeon, K. Do, An in-depth analysis on trafficflooding-attacks detect. and sys. using DM technique. J SystArchitect 59-10, 1005–1012 (2013) [CrossRef]
18. K. Levent, S Sarkani, M. Thomas, A network IDS basedon a Hidden NB multiclass-classifier. Expert SystAppl 13492–13500 in 39-18 (2012) [CrossRef]
19. R. Samaneh, C Lam, H. Philip, Evolving stat. rule sets for networkintrusion detec. App.Soft Compt. 348–359, 33 (2015) [CrossRef]
20. E. Heba, H. Aboul, B. Soumya, K. Taihoon, Proceeds. of first InternationalConference on Adv. in Secu. of Info. and CommunicationNetworks (Sec-Net). Linear correlationbased FS for networkintrusion detectn.model, pp. 240–248 Cairo, Springer (2013) [CrossRef]

AUTHORS PROFILE

Darshan Thakur, received the B.Tech degree in Computer Engineering from Savitribai Phule Pune University, India in 2020. He is currently pursuing Masters in Technology within computer engineering stream from College of Engineering, Pune, India. His topics of interest are information security, ethical hacking, cloud computing and applications of machine learning. He has been a member of IEEE Student Chapter Pune in the year 2018. His research interests include machine learning and information security.

Tanuja Pattanshetti, is currently working as assistant professor at College of Engineering, Pune, India. She instructs in a variety of areas, including big data, software engineering, and cloud computing. Her research interests include Big Data, Data preprocessing frameworks, Software engineering, and Machine Learning. Various research works have been published by her in the research domains like Data preprocessing and Big data. She is an member of Indian Society for Technical Education (ISTE).