International Journal of Information Security and Privacy
Volume 14 • Issue 1 • January-March 2020

An Improved Intrusion Detection System to Preserve Security in Cloud Environment

Partha Ghosh, Netaji Subhash Engineering College, MAKAUT, Kolkata, India
Sumit Biswas, Tata Consultancy Services Ltd, Mumbai, India
Shivam Shakti, Netaji Subhash Engineering College, MAKAUT, Kolkata, India
Santanu Phadikar, Maulana Abul Kalam Azad University of Technology, Kolkata, India

ABSTRACT

Cloud computing, also known as on-demand computing, provides different kinds of services for the users. As the name suggests, its increasing demand makes it prone to various intruders affecting the privacy and integrity of the data stored in the cloud. To cope with this situation, intrusion detection systems (IDS) are implemented in the cloud. An effective IDS constitutes of less time-consuming algorithm with less space complexity and higher accuracy. To do so, the number of features are reduced while maintaining minimal loss of information. In this paper, the authors have proposed a model by which the features are selected on the basis of mutual information gain among correlated features. To achieve this, they first group the features according to the correlativity. Then from each group, the features with the highest mutual information gain in their respective groups are selected. This led them to a reduced feature set which provides quick learning and thus produces a better IDS that would secure the data in the cloud.

KEYWORDS

Cloud Computing, Core Cluster, Feature Selection, Intrusion Detection System (IDS), Mutual Information (MI)

INTRODUCTION

Cloud computing is a widespread term for the transportation of hosted services using the Internet. Cloud computing has evolved as one of the most vital dimension of the modern software industry by making a transition from computing-as-a-product to computing-as-a-service (Murugesan, 2011). Instead of setting up a physical infrastructure, Cloud allows us to have the luxury of using applications, software, platforms etc. as a service and one has to pay only for the resources he consumes (Singh & Jangwal, 2012). Since in a Cloud Environment data arrives from different heterogeneous sources therefore understanding the associative vulnerabilities is the foremost job to do (Grobauer, Walloschek, & Stöcker, 2011) and then, to provide a way to maintain the integrity, confidentiality and availability of the incoming and outgoing data. Hamlen et al. (Hamlen, Kantarcioglu, Khan, & Thuraisingham, 2010) in their work have discussed the various security issues of the Cloud. IDS is one such solution that provides data security to the Cloud Environment. Based on deployment, IDS have two models, Host Based IDS(HIDS) and Network Based IDS(NIDS). HIDS attempts to recognize unauthorized, abnormal behaviors on a specific device (Hu, 2010). HIDS uses both Anomaly Based and Misuse Based Detection Techniques and plays a very compliant role in identifying, logging records and alerting the admin if there is any security issue. Whereas NIDS completely works on Network

DOI: 10.4018/IJISP.2020010105

traffic. It captures Ethernet Packets and scans it in real time to decide whether it is an attack or not (Mukherjee, Heberlein, & Levitt, 1994). The number of unnecessary generated alerts in Anomaly Based IDS which causes high false alarm can be reduced as demonstrated by Hacini et al. (Salima Hacini, Zahia Guessoum, 2013) .

As the network traffic is huge in size so the analysis of packets in real time is too time-consuming phenomenon, hence for better performance of IDS it is incorporated with various data mining algorithms extensively (Yanjie, 2015). For further enhancement in the performance pre-processing of data becomes inevitable which reduces dimensions quite significantly (Said, Stirling, Federolf, & Barker, 2011). Feature Selection is one of the most widely used pre-processing technique which eliminates irrelevant and homogeneous features from a given feature set (Mladeni, 2006). Another pre-processing technique is Clustering which helps to eliminate outliers, noise and group similar kind of objects. Objects can be either instances or features (Kryszkiewicz & Skonieczny, 2005). For the experimental purpose the authors have used NSL-KDD dataset for training and testing purpose. In this paper, initially authors have designed a fully connected weighted graph of features, where each node represents a feature. Then Core Clusters are created by removing inconsistent edges. Later, the relevant features which have high Mutual Information Values, are selected from each core in order to get the Relevant Feature Set (RFS). Using the above mentioned methods the authors have proposed an Anomaly Based Intrusion Detection System.

RELATED WORK

Cloud has been an inseparable part of modern day technology because of its fascinating storage and computing capability. Therefore various conventional services such as Messaging Services, Social Networking Services are shifting towards Cloud Platform. Shawish and Salama (Shawish & Salama, 2014) gave a overview of Cloud's anatomy, characteristics and architecture. They also covered a detailed comparison between Cloud Based Services and Existing Services. Though Cloud is very flexible but it is quite vulnerable to various kinds of attacks. To overcome all these data security issues various IDSs are required. A brief introduction to IDS was proposed by Mohamed et al. (A. Mohamed, Idris, & Shanmugum, 2012). In their work they reviewed IDS, pointed out those issues that appeared during implementation of IDS and the restrictions in the research field in IDS. Kumar et al. (B. S. Kumar et al., 2001) provided an in depth description of IDS model and the various types of intrusion in the system and their corresponding prevention techniques. Lombardi and Di Pietro (Lombardi & Di Pietro, 2011) proved how Virtualization can be implemented to increase security in Cloud. They proposed a novel architecture, Advanced Cloud Protection System(ACPS) that can fruitfully audit the integrity of the Cloud Environment. Denning (Denning, 1987) developed a general purpose IDS framework which was system as well as environment independent and consolidated the fact that, security breaches can be identified by monitoring system's log of unusual patterns. Kholidy and Baiardi (Kholidy & Baiardi, 2012) outlined a framework to work out the inadequacy of IDS model. They incorporated both Knowledge-Based and Behavior-Based techniques to improve the overall attack handling capability. Later on, for the betterment of IDS performance in Cloud, several data mining techniques were also introduced. Lee and Stolfo (Lee & Stolfo, 2000) suggested a model which was based on data mining algorithms. The model used Classification, Meta-Classification, Association and Frequent Rule to generate frequent pattern from audit log to detect anomalies. The result displays that the model is as good as those systems which were manual knowledge approach driven. Mohamed et al. (S. Mohamed, Mohamed, & Mokhtar, 2017) proposed an IDS model using a hybrid approach of K-Means and Sequential Minimal Optimization (SMO) Classification. They apply the approach on NSL-KDD dataset and the result shows that it brings down the false alarm rate quite magnificently and achieves higher accuracy. Few Denials of Service(DoS) attacks can bypass both the application and operating system layer which imposes serious threats. That's why Tao et al. (Tao, Yang, Peng, & Li, n.d.)proposed a HIDS which shows better detection rate of DoS

intrusions. Gupta et al. (Gupta, Singhal, & Malik, 2016) developed a NIDS based on different data mining approaches involving Linear Regression and K-Means clustering to automatically discover the classification rules. Kumar et al. (G. Kumar, Saha, Singh, & Rai, 2018) also proposed a NIDS model where they used snooping agents and honeypot which demonstrates how different attack sequences affects the network performance such as throughput, network load, retransmission etc. Using NSL-KDD dataset, a comparative analysis is done by them. Rao et al. (Rao, Damodaram, & Charyulu, 2012) further proposed Modified and Hashed K-Means approach which overcomes the drawbacks of K-Means method. Their model is deployed on the KDD99 dataset and produces satisfactory improvement in the efficiency of intrusion detection. A comparative study of Fuzzy C-Means(FCM) and K-Means was done by Nadiammai and Hemalatha (Nadiammai & Hemalatha, 2012). The result section shows that FCM outperforms K-Means in both correctness and speed criterions. Ghosh et al. (Ghosh, Mandal, & Kumar, 2015) recommended an efficient IDS model that merged up both multi-threaded NIDS and HIDS. The system used to capture, scan packets from network and report to the admin. The model was good enough to handle massive data flow, scan them and produce report by integrating both Misuse and Anomaly Detection. Since IDS works with a large set of data therefore to further enhance the performance of the system various data pre-processing techniques are introduced, such as Feature Selection. Feature Selection is one of the major parts in Machine Learning Domain. Ghosh et al. (Ghosh, Debnath, Metia, & Dutta, 2014) proposed a multilevel Hybrid IDS model. They used K-Nearest Neighbor (KNN) as binary classifier and selected relevant features from Feature Set before classification to reduce the training time. The result shows that they got better classification accuracy with their Hybrid Model. Ganapathy et al. (Ganapathy et al., 2013) did a survey of various Intelligent Feature Selection and Classification Techniques in IDS. They proposed two new Intelligent Approaches in their paper, Rule Based Attribute Selection and Rule Based Advanced Multiclass SVM. Dreiseitl and Osl (Dreiseitl & Osl, 2009) outlined a Hybrid model combining both Wrapper and Filter approaches to select relevant features from a large set of features. The result shows that the model outperformed the Filter Methods. Sharmin et al. (Sharmin, Ali, Khan, & Shoyaibl, 2017) presented an IDS model where they implemented discretization and feature selection based on Mutual Information. They implemented their model on several datasets and the experimental results showed better performance. An IDS model with a combination of K-means algorithm, which was based on Cosine Similarity as distance metric and Information Gain (IG) was proposed by Dubey et al. (Dubey, Saxena, & Shrivas, 2016). Cosine Similarity was used to cluster the features which are highly similar and IG was used to select most useful features from each cluster. They used Naive Bayes, KNN and CART Classifiers to obtain Classification accuracy and compared it with filter-based feature selection technique which led them to observe improved efficiency along with substantially reduced number of features. The above-mentioned works help the authors to build a brief idea about Cloud, IDS, Feature Selection Techniques and they have implemented Feature Selection Technique in their proposed model by incorporating Core Clusters and Mutual Information (MI). The authors generate the Core Clusters based on the Mutual Information Gain values such that the Relevant Feature Set obtained has the most diverse and non-redundant features which facilitate quick training of classification models on the NSL-KDD dataset with improved accuracy.

PROPOSED MODEL

Technology is becoming much more sophisticated after every passing day, so does the attacker. Therefore, to prevent all these attackers and provide data security, an efficient IDS model is the finest option. Now, to increase the efficiency of an IDS model, time complexity and space complexity of the training dataset are needed to be reduced. Feature Selection, is one of the ways to lessen this complexity, which can be achieved by removing irrelevant features from the dataset. During the last decade, the field of Feature Selection in Cloud Environment has been thoroughly inspected by the researchers and several IDS models have been proposed accordingly. Based on the advantages and

drawbacks of these existing models, the authors have proposed an Anomaly Based IDS model in this paper.

In this paper, authors have used the NSL-KDD benchmark dataset for evaluation of the proposed model. NSL-KDD dataset contains four components: "KDDTrain+", "20%KDDTraining+", "KDDTest+" and "KDDTest21". "KDDTrain+" and "KDDTest+" have been used for training and testing purposes which involves 125793 and 22544 tuples respectively.

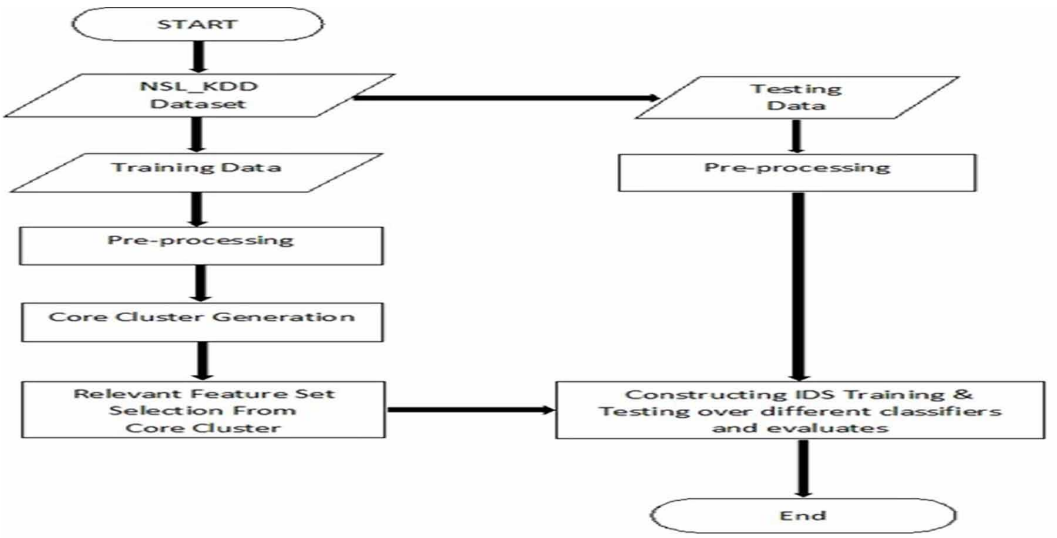
Authors proposed a feature selection algorithm to obtain relevant features using the concepts of Graph theory and correlation between features. Mutual Co-Relation is a function that defines the relationship between random variables, based on various metrics between those variables (Cominetti et al., 2010). Authors reckon the RFS as a set of features which are highly related to most of the features of the dataset and have the ability to efficiently constitute both the dataset class distribution and the original dataset. The degree of association between attributes is visualized through a fully connected weighted graph, where edges represent the pairwise correlation values between nodes. Here in the graph each node represents a feature and unlike most of the existing wrapper-based feature selection algorithms, authors have proposed a method that uses Filter-Based Feature selection.

Authors' algorithm consists of two phases. First phase is called Core Cluster Generation Phase, where the Co-Relation between features is calculated in order to find out how closely a feature is related with others and to group them into several connected components known as Core Clusters, which are derived from the fully connected graph G(V,E). RFS selection is the second phase where each core is analyzed and all the features in that core are assigned with Mutual Information(MI) values based on a class label. Then, the algorithm extracts the most diverse and non-redundant features in order to maintain the diversity of the RFS. Figure 1 sketches all the steps of the proposed IDS Model.

Core Cluster Generation

In Core Cluster generation authors calculate the Pearson-Correlation Coefficient (PCC) of all pairs of features (Peng, Long, & Ding, 2005). Usually a feature set is considered to be good if it contains features with high correlation to the class and there is low redundancy amongst them. In their approach the authors have emphasized on the correlation values of the attributes to determine how close they are to each other. The value means two things, firstly, highly correlated features tend to lie in the

Figure 1. Flowchart of the proposed model



same Core Cluster and secondly, features with relatively less correlation will certainly belongs to different Core Clusters.

In authors' proposed algorithm Effectiveness of Relationship (EOR) between two features is decided on the magnitude of Pearson value. The formula for Pearson Correlation is given below,

$$\rho = \frac{\sum\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sqrt{\sum\left(x_i - \overline{x}\right)^2 \sum\left(y_i - \overline{y}\right)^2}}, -1 \le r \le 1 \tag{1}$$

Where X and Y are two features and $\forall\, x_i \in X$ and $\forall\, y_i \in Y$ are the dimensions of the features. The score is calculated for the pair of X and Y using equation 2 is given below,

$$PCC\left(X,Y\right) = \begin{cases} \left|\rho\right| \ if\, X \neq Y \\ 0\, otherwise \end{cases} \tag{2}$$

Successively for each pair of features, the correlation values are found out and the graph is converted into a completely weighted graph. Thereafter, authors obtain a threshold value of PCC - Thresh_PCC, which is used to remove all those edges which have less PCC values than that of the threshold in order to avoid ineffective relationships.

Iteratively following this, several separate connected components are created, known as Core Clusters where in each core there reside features that are highly correlated.

In a nutshell, the algorithm for Core Cluster generation is written in Algorithm 1,

Algorithm 1: Core Cluster Generation

Input: Conditional attributes P represented as Vertex in fully connected graph G(V,E)

Output: Several connected components named Core Clusters C(C1,C2,..,Cn) originated from graph G(V,E)

1: Take PCC = matrix[|P|][|P|], indexed by X, Y ϵ P
2: for all X, Y ϵ P do
3: if X ≠ Y then
4: Compute PCC (X, Y) //using equations (1) and (2) //
5: W(X,Y) = EOR(PCC(X,Y)) //Weight of edge (X,Y)//
6: end if
7: end for
8: for all X,Y ϵ P do
9: if EOR(PCC(X,Y)) < Thresh_PCC value then
10: W(X,Y)=0 (Remove that edge (X,Y))
11. end if
12:end for

Relevant Feature Set (RFS) Generation

This phase processes the Cores produced by the Core Cluster Generation Algorithm in order to obtain the RFS. Authors select each Core and analyze all the features inside it very closely. Since a core may consist of one or more than one features, therefore to select the most useful one, authors have used Mutual Information (MI) criteria. Here MI values are calculated as MIinfo(f,cl), where f is a

feature and cl is the corresponding class. MI evaluates how much information presence or absence of a feature f, contributes towards achieving the correct classification decision on cl.

In the proposed model, all the features of a particular Core have been sorted in a decreasing manner based on their MI values. This serves two purposes, firstly, the feature at the top with the maximum MI value classifies each record better than the others and secondly, feature having the least MI value hardly has any vital role to play in correct classification. After analyzing each and every Core the top features from each core is extracted and added to the Feature Set(FS). Here a scenario may occur where a Core may contain only one feature and therefore irrespective of the MI value of the feature it will be extracted and added to the FS. Henceforth, to make our proposed FS relevant, a filtration process is applied. Authors obtain a threshold value of MI, named Thresh_MI. Thresh_MI value helps to remove all those features from FS whose MI values are less than the threshold. Following this, it can create an RFS which contains only the relevant and useful features that has the maximum contribution to make correct classification decision.

The overall algorithm of the proposed model is given below in Algorithm 2,

Algorithm 2: Algorithm of the Proposed Model

Input: Training Dataset.

Output: Relevant Feature Set (RFS).

1. Construct a fully connected graph with features as nodes.
2. For every pair of nodes assign the pairwise correlation values as weights.
3. For all edges in the graph G(V, E)
4. Traverse an edge E_i
5. If the co-relation value of the edge E_i is less than the given Threshold value then,
6. remove that edge
7. End if
8. Select a new edge(i++)
9. Consider each connected component as Core derived from the fully connected graph.
10. For each Core sort all the features based on their Mutual Information Score in a decreasing order.
11. Select the top feature from each core and add it to the FS.
12. Filtration is done by removing those features from FS whose MI value is lower than the given Threshold value and the RFS is found.

Figure 2 represents all the steps that are required to generate the RFS.

Execution of the proposed model is presented here with the help of an example.

Initially, F (set of features) = {f1,f2,f3,….,f6} which consist of six features. FS = { Φ }, C(Core Clusters)= { Φ }, RFS={ Φ } and all the six features are represented as nodes in a fully connected weighted graph G(V,E), where the weights represents the correlation between two features.

Figure 3 represents the step of generating Core Clusters from the fully connected weighted graph G(V,E).

At first the graph G(V,E) is fed to the Core Cluster Generation Algorithm. Now the algorithm will iteratively remove edges from the graph G where the weight is less than the given Thresh_PCC value. Assume after the algorithm terminates there are three connected components or Core Clusters C1, C2 and C3 respectively where C1= {f1, f5, f6}, C2= {f2, f4} and C3= {f3}. Thereafter the output of Algorithm 1 is fed to the next step, RFS generation. Here authors calculate the Thresh_MI and MI values for all features belong to each and every Core Cluster. After that sort those in decreasing order based on MI values. Assume for C1 it is {f1>f6>f5}, C2= {f2>f4} and C3= {f3}. Top feature from each Core is extracted i.e. f1,f2,f3 and added to the FS. Finally filtration is done on FS by removing features whose MI value is less than the Thresh_MI and assume here f3 is removed and the final RFS which consist of {f1,f2}.

Figure 2. Flowchart of RFS generation

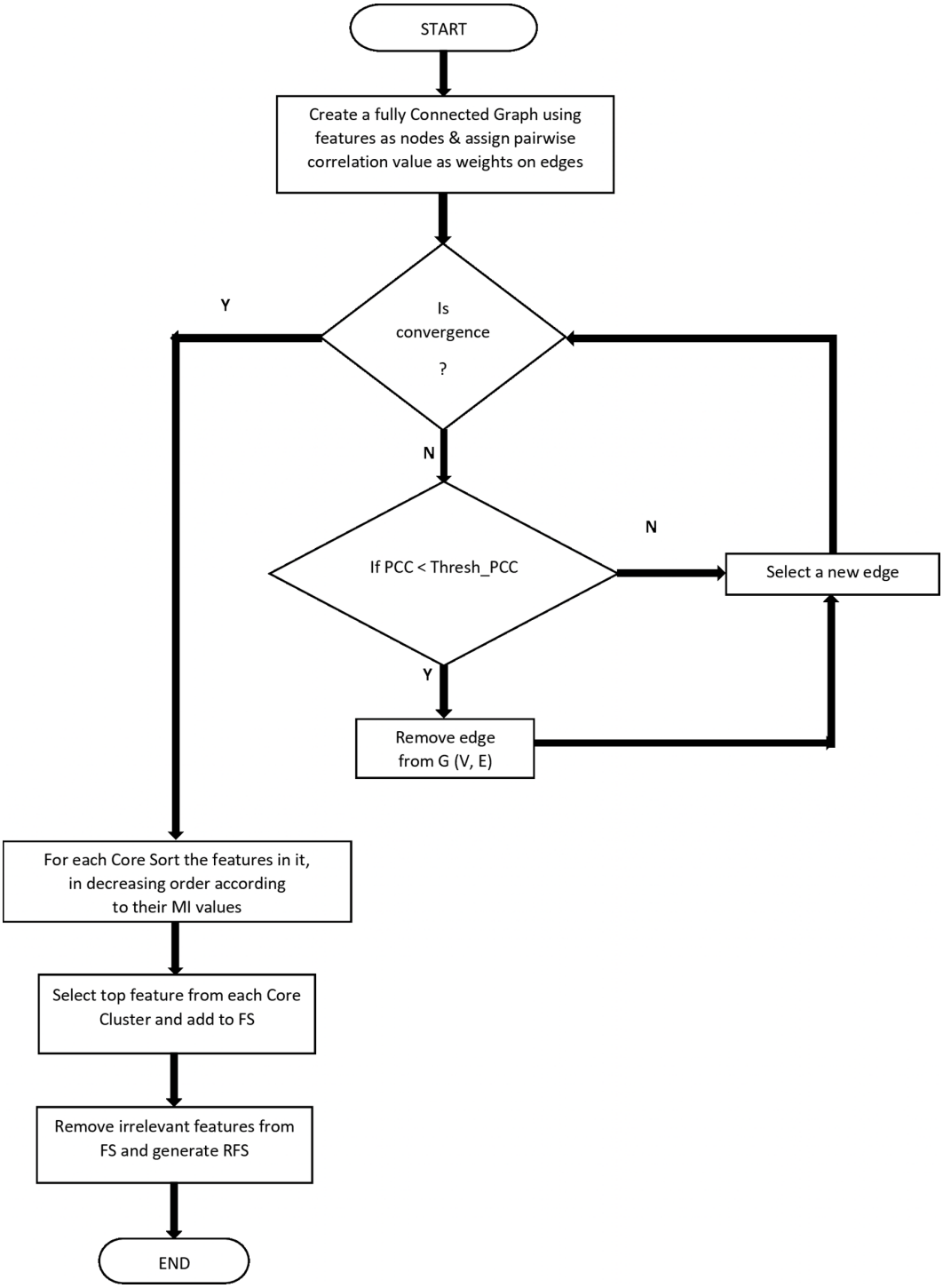
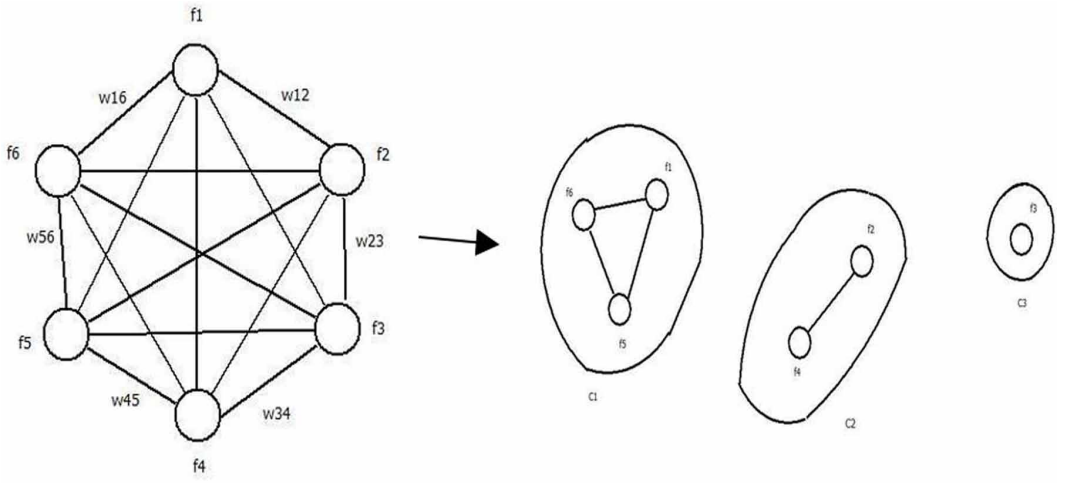


Figure 3. Creation of core clusters from fully connected weighted graph



The proposed model involves the creation of a RFS and later it is used for classification and the results are stored. Thereafter Logistic Regression, Random Forest and AdaBoost classifiers are applied on the datasets for classification purpose and the results are stored in a tabular form. Results obtained from these classifiers are compared with the proposed model's outcome and it shows that the proposed model yields far better classification accuracy than these classifiers. The algorithm distinctly improves the efficiency of IDS by minimizing the total number of features to be considered while stamping a record as normal or attack.

RESULT AND ANALYSIS

In this paper the authors have presented how the number of features required to train an IDS can be greatly reduced, bringing down the cost and time of learning by a significant margin. Efficient feature selection is a crucial stage before training any machine learning model and more so, when Cloud security is the concern.

They have applied their feature selection model on the NSL-KDD-Train+ dataset and found that the number of selected features was only 13 which is significantly less than the original number of features in the dataset which stands at 41.

First, the authors have formed Cores from the highly correlated features and have plotted the correlativity between the features, as given in Figure 4.

Then the authors have used these selected features to train some popular classifier models and compared the accuracy score of the classifier when tested on the NSL-KDD-Test+ against the accuracy score produced by the same classifier model when trained with the complete feature set (Tables 1 and 2). The training time of the models was found in a Jupyter Notebook using (Anaconda) Python 3.5.2 on a 6GB machine running Ubuntu 16.04.

The reduced feature set which is produced by the proposed algorithm is used to train three popular classification algorithms, namely - Logistic Regression, Random Forest and AdaBoost on the NSL-KDD dataset for intrusion detection. The time required for training the models and the accuracy achieved by the models on the test dataset is recorded. To measure the comparative performance, the aforementioned models are again trained with the complete feature set present in the NSL-KDD dataset and as before, the training time and classification accuracy is recorded. It is

Figure 4. Correlativity of the features

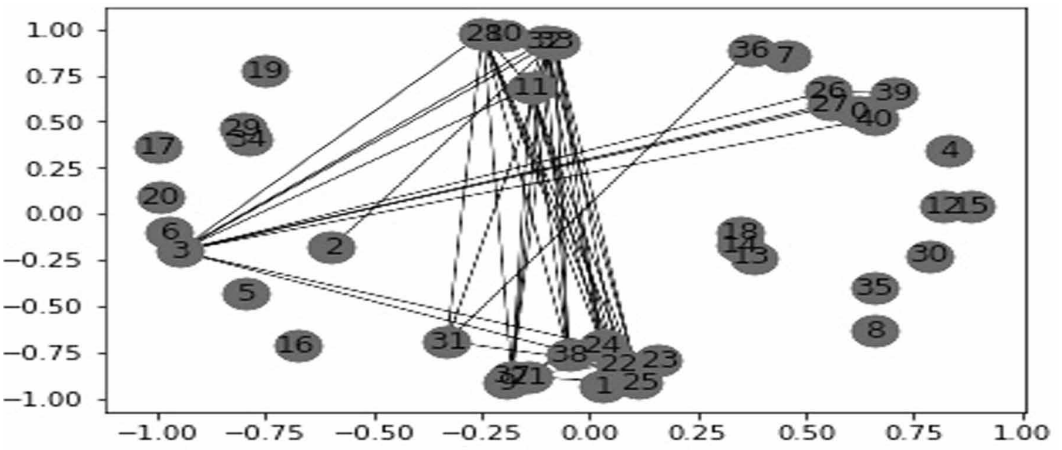


Table 1. Confusion matrix for logistic regression trained on complete feature set (trained by KDDTrain+)

	Attack	Normal
Attack	7910	4923
Normal	635	9076

Table 2. Confusion matrix for logistic regression trained on reduced feature set (trained by KDDTrain+)

	Attack	Normal
Attack	9104	3729
Normal	747	8964

thus clearly observed that the training time for the classifier algorithms is always less while training on the reduced feature set (Figures 5 and 6). At the same time, when the classifier algorithms are trained on the reduced feature set, they mostly achieve better classification accuracy as compared to when they are trained on the complete feature set. Further, memory consumption is decreased due to reduced number of features in the training dataset.

The above tabulated observations (Table 3) lead the authors to claim that the proposed algorithm helps in reducing the time taken and the memory required to train the classifier models while maintaining, or improving, the accuracy of classification for the models.

CONCLUSION

From the results and analysis presented by the authors in this paper, it is very easy to conclude that the proposed feature selection model clearly outperforms the scenario of training the classifier algorithms with complete feature set on the parameters - speed and accuracy. Such performance is highly desirable in a Cloud based IDS where a small delay in learning and hence, identifying malicious intrusions can lead to catastrophic events for organizations or individuals. The proposed model in this paper displays a novel utilization of feature correlation. It establishes that it is possible to greatly reduce

Figure 5. Comparison of different Classifiers (trained using complete vs reduced train set)

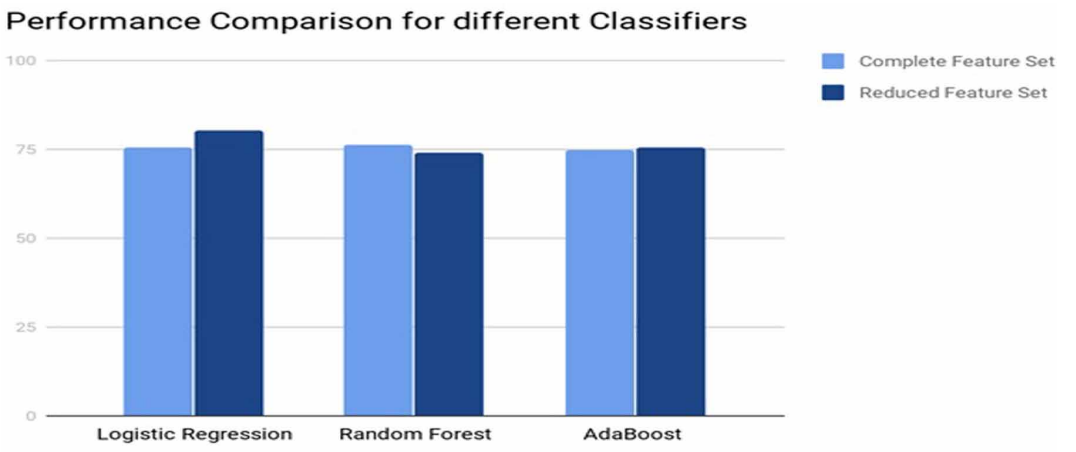
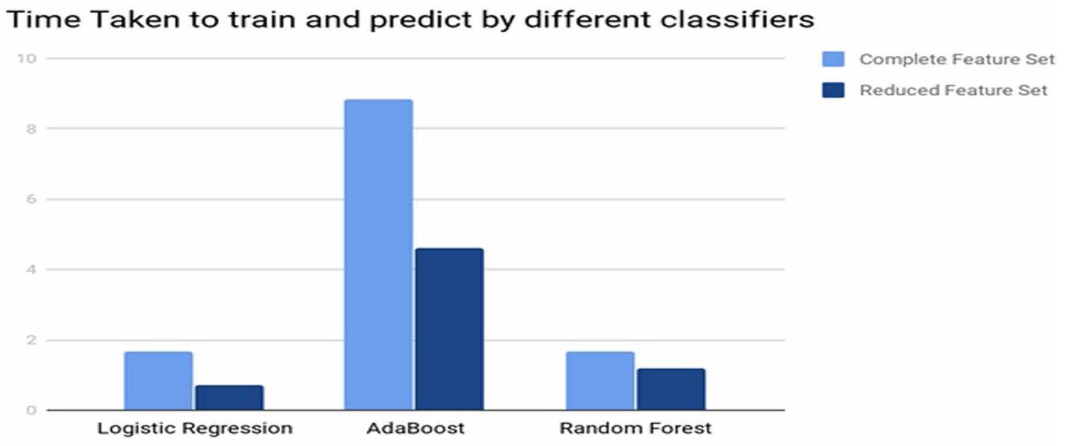


Figure 6. Comparison of training time for different classifiers



the feature set on the basis of feature correlation in the NSL-KDD dataset, without loss in accuracy and at the same time bringing down the training time of the classifier. The experimental results also depicts that the proposed algorithm is fit for being a proficient IDS in the Cloud Environment.

Table 3. Comparative performance analysis using different classifiers

	Logistic Regression		Random Forest		AdaBoost	
	Complete Feature set	**Reduced Feature set**	**Complete Feature set**	**Reduced Feature set**	**Complete Feature set**	**Reduced Feature set**
TP	7910	9104	7737	7708	7871	7693
TN	9076	8964	9452	8968	8987	9366
FP	635	747	259	743	724	345
FN	4923	3729	5096	5125	4962	5140
TPR	61.63796462245772	70.94210239226993	60.28987765915998	60.063897763578275	61.3340606249513	59.947011610691185
TNR	93.46102358150551	92.3076923076923	97.33292142930698	92.3488827103285	92.54453712285037	96.44732777262898
FPR	6.538976418494485	7.692307692307693	2.667078570693022	7.651117289671504	7.455462877149628	3.552672227371019
FNR	38.36203537754228	29.05789760773007	39.71012234084002	39.936102236421725	38.6659393750487	40.052988389308815
Accuracy	75.3459006387509	80.14549325762953	76.24645138396026	73.97090134847409	74.77821149751597	75.66980127750177
Time (sec)	1.6670534589993622	0.7051696099988476	1.687639625999509	1.202354323999316	8.840211908000128	4.619828058999701

REFERENCES

Cominetti, O., Matzavinos, A., Samarasinghe, S., Kulasiri, D., Liu, S., Maini, P. K., & Erban, R. (2010). DifFUZZY: A fuzzy clustering algorithm for complex datasets. *International Journal of Computational Intelligence in Bioinformatics and Systems Biology*, *1*(4), 402. doi:10.1504/IJCIBSB.2010.038222

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering*, *13*(2), 222–232. doi:10.1109/TSE.1987.232894

Dreiseitl, S., & Osl, M. (2009). Feature Selection Based on Pairwise Classification Performance. In *International Conference on Computer Aided Systems Theory*. Springer. doi:10.1007/978-3-642-04772-5_99

Dubey, V. K., Saxena, A. K., & Shrivas, M. M. (2016). A Cluster-Filter Feature Selection Approach. In *International Conference on ICT in Business Industry & Government (ICTBIG)*. IEEE. doi:10.1109/ICTBIG.2016.7892637

Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, L., & Kannan, A. (2013). Intelligent feature selection and classification techniques for intrusion detection in networks: A survey. *EURASIP Journal on Wireless Communications and Networking*, (1): 1–16. doi:10.1186/1687-1499-2013-271

Ghosh, P., Debnath, C., Metia, D., & Dutta, R. (2014). An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment. *IOSR Journal of Computer Engineering*, *16*(4), 16–26. doi:10.9790/0661-16471626

Ghosh, P., Mandal, A. K., & Kumar, R. (2015). An Efficient Cloud Network Intrusion Detection System. In *Advances in Intelligent Systems and Computing* (Vol. 339, pp. 143–152). New Delhi: Springer India; doi:10.1007/978-81-322-2250-7

Grobauer, B., Walloschek, T., & Stöcker, E. (2011). Understanding cloud computing vulnerabilities. *IEEE Security and Privacy*, *9*(2), 50–57. doi:10.1109/MSP.2010.115

Gupta, D., Singhal, S., & Malik, S. (2016). Network Intrusion Detection System Using various data mining techniques. In *International Conference on Research Advances in Integrated Navigation Systems (RAINS - 2016)* (pp. 1–6). IEEE. doi:10.1109/RAINS.2016.7764418

Hacini, S., Guessoum, Z., & Cheikh, M. (2013). False Alarm Reduction Using Adaptive Agent-Based Profiling. *International Journal of Information Security and Privacy*, *7*(4), 53–74. doi:10.4018/ijisp.2013100105

Hamlen, K., Kantarcioglu, M., Khan, L., & Thuraisingham, B. (2010). Security Issues for Cloud Computing. *International Journal of Information Security and Privacy*, *4*(2), 36–48. doi:10.4018/jisp.2010040103

Hu, J. (2010). Host-based anomaly intrusion detection. *Handbook of Information and Communication Security*, 235–255. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-04117-4_13

Kholidy, H. A., & Baiardi, F. (2012). CIDS: A framework for intrusion detection in cloud systems. *Proceedings of the 9th International Conference on Information Technology, ITNG 2012*, 379–385. doi:10.1109/ITNG.2012.94

Kryszkiewicz, M., & Skonieczny, Ł. (2005). Faster Clustering with DBSCAN. *Intelligent Information Processing and Web Mining*, 605–614. 10.1007/3-540-32392-9_73

Kumar, B. S., Sekhara, T. C., Raju, P., Ratnakar, M., Baba, S. D., & Sudhakar, N. (2001). Intrusion Detection System- Types and Prevention. *International Journal of Computer Science and Information Technologies*, *4*(1), 77–82.

Kumar, G., Saha, R., Singh, M., & Rai, M. K. (2018). Optimized Packet Filtering Honeypot with Snooping Agents in Intrusion Detection System for WLAN. *International Journal of Information Security and Privacy*, *12*(1), 53–62. doi:10.4018/IJISP.2018010105

Lee, W., & Stolfo, S. J. (2000). A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security* (Vol. 3). 10.1145/382912.382914

Lombardi, F., & Di Pietro, R. (2011). Secure virtualization for cloud computing. *Journal of Network and Computer Applications*, *34*(4), 1113–1122. doi:10.1016/j.jnca.2010.06.008

Mladeni, D. (2006). Feature Selection for Dimensionality Reduction. *Subspace*. *Latent Structure and Feature Selection*, *3940*, 84–102. doi:10.1007/11752790_5

Mohamed, A., Idris, N., & Shanmugum, B. (2012). A Brief Introduction to Intrusion Detection System. *Trends in Intelligent Robotics,* 263–271. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-35197-6_29

Mohamed, S., Mohamed, A., & Mokhtar, R. A. (2017). Anomaly Detection Approach using Hybrid Algorithm of Data Mining Technique. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, Sudan.

Mukherjee, B., Heberlein, L. T., & Levitt, K. N. (1994). Network intrusion detection. *IEEE Network*, *8*(3), 26–41. doi:10.1109/65.283931

Murugesan, S. (2011). Cloud computing gives emerging markets a lift. *IT Professional*, *13*(6), 60–62. doi:10.1109/MITP.2011.94

Nadiammai, G. V., & Hemalatha, M. (2012). An evaluation of clustering technique over intrusion detection system. *Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12*, 1054–1060. doi:10.1145/2345396.2345565

Peng, H., Long, F., & Ding, C. (2005). Feature Selection Based on Mutual Information :Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. doi:10.1109/TPAMI.2005.159 PMID:16119262

Rao, M. V., Damodaram, A., & Charyulu, N. C. B. (2012). Algorithm for Clustering with Intrusion Detection Using Modified and Hashed K – Means Algorithms. In *Advances in Computer Science, Engineering & Applications* (pp. 737–744). Berlin: Springer. doi:10.1007/978-3-642-30111-7_70

Said, D., Stirling, L., Federolf, P., & Barker, K. (2011). Data preprocessing for distance-based unsupervised Intrusion Detection. *2011 9th Annual International Conference on Privacy, Security and Trust, PST 2011*, 181–188. doi:10.1109/PST.2011.5971981

Sharmin, S., Ali, A. A., Khan, M. A., & Shoyaibl, M. (2017). *Feature Selection and Discretization based on Mutual Information*. IEEE. doi:10.1109/ICIVPR.2017.7890885

Shawish, A., & Salama, M. (2014). Inter-cooperative Collective Intelligence. *Techniques and Applications*, *495*, 39–67. doi:10.1007/978-3-642-35016-0

Singh, S., & Jangwal, T. (2012). Cost breakdown of Public Cloud Computing and Private Cloud Computing and Security Issues. *International Journal of Computer Science & Information Technology*, *4*(2), 17–32. doi:10.5121/ijcsit.2012.4202

Tao, R., Yang, L., Peng, L., & Li, B. (n.d.). A Host-Based Intrusion Detection System Using Architectural Features to Improve Sophisticated Denial-of-Service Attack Detections. *Optimizing Information Security and Advancing Privacy Assurance*, 18–31. 10.4018/978-1-4666-0026-3.ch002

Yanjie, Z. (2015). Network Intrusion Detection System Model Based on Data Mining. In *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS* (Vol. 9, pp. 359–370). IEEE. doi:10.1109/SNPD.2016.7515894

Partha Ghosh is an Assistant Professor of Information Technology at Netaji Subhash Engineering College, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India. He has done M.Tech. in Computer Science and Engineering from Calcutta University in 2003. His research interests include Cloud Computing, Machine Learning, Intrusion Detection System, Computer Networks and Security.

Sumit Biswas pursued his B.Tech in Information Technology from Netaji Subhash Engineering College, Maulana Abul Kalam Azad University Of Technology, Kolkata in 2017. He is currently a part of Tata Consultancy Services Limited, Kolkata as a Developer. His research area includes Artificial Intelligence, Machine Learning, Cloud Computing, Analytics and Intrusion Detection System.

Shivam Shakti is a student of B.Tech. in Information Technology at Netaji Subhash Engineering College, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India. His interests include Cloud Computing and Machine Learning.

Santanu Phadikar is an Associate Professor and Head of the department of Computer Science and Engineering at Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India. He has done M.Tech. in Computer Science and Engineering from Calcutta University in 2003. He pursued his Ph.D. from Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India in 2013. His research area includes Machine Learning, Intrusion Detection System, Soft Computing and Cloud Computing.