International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 10 | Oct 2023 www.irjet.net p-ISSN: 2395-0072

A PHASED APPROACH TO INTRUSION DETECTION IN NETWORK

Pankaj Kumar[1], Mr. Sambhav Agrawal[2]

[1]M.Tech, Computer Science and Engineering, SR Institute of Management & Technology, Lucknow, India
[2]Associate Professor, Computer Science and Engineering, SR Institute of Management & Technology, Lucknow

---

Abstract - *Network security is of paramount importance in today's digital landscape, where cyber threats continue to evolve and become more sophisticated. Intrusion detection systems (IDS) play a crucial role in identifying and mitigating these threats, but they face challenges in detecting attacks accurately and efficiently. In this research paper, we propose a multi-stage intrusion detection approach that aims to enhance network security by effectively detecting and classifying various types of intrusions.*

The proposed approach consists of multiple stages, each designed to address specific aspects of intrusion detection. The first stage focuses on pre-processing and feature extraction, where relevant network traffic data is collected and transformed into meaningful features for subsequent analysis. Machine learning algorithms are applied in the second stage for building models capable of recognizing normal network behavior and identifying potential intrusions. These models are trained using a comprehensive dataset that encompasses various types of attacks and normal network traffic.

To further improve the accuracy of intrusion detection, the third stage incorporates a rule-based system that applies specific rules and thresholds to the output of the machine learning models. This allows for more fine-grained classification of network activity and reduces false positives. Additionally, the proposed approach leverages anomaly detection techniques in the fourth stage to identify novel or unknown attacks that may not conform to predefined patterns.

Key Words: GBBK+, k-point+, Multi-Stage Intrusion Detection System, outlier detection, SVM.

1. INTRODUCTION

Network security is a critical concern in today's interconnected world. With the increasing sophistication of cyber threats, effective intrusion detection systems (IDS) are crucial for identifying and mitigating potential attacks. Traditional IDS often face challenges in accurately and efficiently detecting various types of intrusions. To address these limitations, this research paper proposes a multi-stage intrusion detection approach that combines different techniques to enhance network security. The purpose of "A Multi-Stage Intrusion Detection Approach for Network Security" is to propose and develop an advanced intrusion detection methodology that addresses the limitations of traditional single-stage IDS systems. The research aims to enhance network security by effectively detecting and classifying various types of intrusions. The multi-stage approach combines different techniques, including pre-processing and feature extraction, machine learning-based classification, rule-based systems, and anomaly detection, to achieve more accurate and efficient intrusion detection. The goal is to provide network administrators with a robust and comprehensive system that improves the overall defense against cyber threats and aids in the protection of critical network infrastructure. The research paper also aims to evaluate the proposed approach using real-world network traffic datasets and compare its performance with traditional IDS systems, thereby validating the effectiveness and benefits of the multi-stage intrusion detection approach.

Classification of Detection Approaches for Network Security

Detection approaches for network security can be classified into several categories based on their underlying techniques and methodologies. These classifications help to provide a clear understanding of the different approaches and their respective strengths and weaknesses. Here are some common classifications:

Signature-Based Detection

Signature-based detection, also known as misuse detection, relies on pre-defined patterns or signatures of known attacks. It compares network traffic or system behavior against a database of signatures and raises an alert if a match is found. While signature-based detection is effective at detecting known attacks, it may struggle with detecting novel or unknown attacks for which no signature exists.

Anomaly-Based Detection

Anomaly-based detection focuses on identifying deviations from established patterns of normal behavior. This approach creates a baseline of normal network or system behavior and alerts on any activity that significantly deviates from the baseline. Anomaly detection is valuable for detecting novel attacks that do not match known signatures. However, it can be challenging to accurately define the normal behavior baseline and distinguish between genuine anomalies and legitimate variations in network activity.

demonstrated that both unsupervised anomaly-based detection and the identification of new assaults without any prior knowledge of labeled training data have the potential to be successful solutions to this problem. Moreover, both of these approaches do not require the presence of a human supervisor. Unsupervised and supervised anomaly detection algorithms are integrated and put to use in a certain manner in order to identify the features of each kind of incursion. This is done by combining the two types of algorithms. This is done with the main goal of obtaining a greater level of performance in mind, thus that should be the driving motivation behind it.

3.1. Simulation

Matlab was used so that a number of simulations of the procedure could be carried out. Using this method, the management of records that include duplicate information or values that are missing may be accomplished. In order to locate the abnormality in the data set that was referred to as the outlier, the procedure that was suggested was put into practise. The dataset is partitioned into several subsets, each of which is established on the basis of the class label that was applied to the record. The final result of the method is created by compiling all of the findings that were acquired from the individual computations of an index for an outlier and combining them together.

4. COMPLEXITY ANALYSIS

The strategy that is recommended makes use of three distinct methods, and their names, in order, are FindNNk(), FindRNNk(), and FindRNOFk(). FindNNk(), FindRNNk(), and FindRNOFk() are the names of some of the methods that may be utilised in this situation (). We are able to calculate the amount of time that will be required to finish the FindNNk() function by first taking into account the distance that exists between n items and then locating the k objects that have the shortest distance between them. This allows us to determine the amount of time that will be required to complete the FindNNk() function. Because of this, we are able to determine the amount of time that will be necessary to complete the function. This function's degree of difficulty is represented by the notation O(n+n), which stands for the phrase "operation plus number." This is as a direct result of the strong connection that exists between these two aspects. The amount of time required to complete the FindRNNK() method is inversely proportional to the total number of distinct items that are examined, which is represented by the notation n*k. The complexity of each item beyond the nth one is represented by the notation O(n*k*n), which stands for "n times k times n." The documentation for the FindRNOFk() function states that it has been given a complexity grade of O, which indicates that it is quite straightforward to use (n). The procedure that was explained has a level of temporal complexity equal to O (n+n+(n*k*n)+n), which is the same thing as O. Because of this, the difficulty of the operation, measured in terms of the amount of time it takes, is equivalent to O. (n2). When viewed from an asymptotic perspective, on the other hand, the level of complexity shown by our approach as well as that of the prior method is same. On the other hand, if our approach is used to a very large number of n, it will provide results that are superior to those obtained by other methods.

5. PERFORMANCE ANALYSIS

According to the research that has been conducted, the outlier distribution is reported to occur anywhere between 10 and 20 percent of the dataset. The number T serves as the criterion for determining how many of the data points are regarded to be anomalous. The findings of this experiment have led us to the conclusion that T should have a value of 0.99, and we got at this conclusion as a consequence of the following: The equation indicates that there is a connection between the threshold value T and the number of outliers, and that this connection has a relationship that is inversely proportional to T. In other words, the relationship is inversely proportional to T.

$$No.\ Of\ Outlier\ \alpha\ \frac{1}{T} \tag{4.3}$$

We constructed a prediction model that was based on the support vector machine so that we could assess whether or not the method that was suggested for spotting outliers is effective (SVM). The dataset was analyzed using this model both before and after the outlier was removed from the image. This allowed for a comparison of the two states of the data. The results on both instances were exactly what was desired and expected. Both the process of training the model and the process of validating the model, which both make use of the training and validation methods, make use of these two distinct iterations of the datasets throughout the whole of both processes. Before the outlier was taken into account, the confusion matrix of the model is shown in Table 1, as it was originally. After taking into account the results of removing the outlier, the confusion matrix of the model can be seen in Table 2. Table 1 displays the model's confusion matrix in its uncorrected form, which means that it does not take into account the outlier.

Table-1: Confusion matrix of the original dataset prior to SVM-based outlier elimination.

11734	9	99.16%
99	13350	99.92%
99.92%	99.27%	99.5713%

In all, there were 25192 cases in the NSLKDD Train20 dataset. The model correctly categorised 25084 of those examples. There is a 98.3 percent chance that this is correct.

As a result of the removal of the outliers, the total number of occurrences was reduced to 24634, and out of that number, only 24530 instances were able to be identified accurately.

Table-2: Confusion matrix of the initial dataset after SVM removal of the outlier.

11475	9	99.17%
95	13055	99.92%
99.91%	99.27%	99.5778%

Tables 1 and 2 each contain a graphical representation of the True Positive (1st cell), the False Negative (2nd cell), the Precision (3rd cell), the False Positive (4th cell), the True Negative (5th cell), the Negative Predicted Value (6th cell), the Sensitivity (7th cell), the Specificity (8th cell), and the Accuracy (9th cell). When compared to one another, the accuracy of the model that does not include the outlier has a much higher level of precision than the accuracy of the model that does include the outlier. To put it another way, the training error of the model that does not include the outlier is much lower than the training error of models that do include the outlier. Because of this, the accuracy of the model is improved when it is applied to data sets that do not include any outliers in comparison to data sets that do have outliers. The dataset ought not to have any extreme values, often known as outliers, since doing so will make it less likely that the model is biassed or that it has been overfit to the data.

In order to create a direct comparison of the amount of time required to compute GBBK and the amount of time required by the suggested technique GBBK+, we carried out 7500 instances at intervals of 500 seconds. A depiction of how well the performance of the total number of instances did may be seen along the X-axis of Figure-1. The Y-axis provides an approximate estimate of the amount of time, measured in seconds, that will be required to finish the execution of the algorithms. This estimate is shown as a range. When compared to the method that was suggested, the graph in Figure-1 illustrates that the amount of time that is required by GBBK constantly increases in proportion to the number of instances that are being processed. This is the case even though the proposed method was intended to reduce the amount of time that is required. When compared to the method that was proposed, this is much different. Even when there is a greater number of instances to handle, the GBBK+ approach completes the task in a shorter amount of time compared to the GBBK algorithm. This is due to the fact that the GBBK+ approach has a reduced number of rounds.

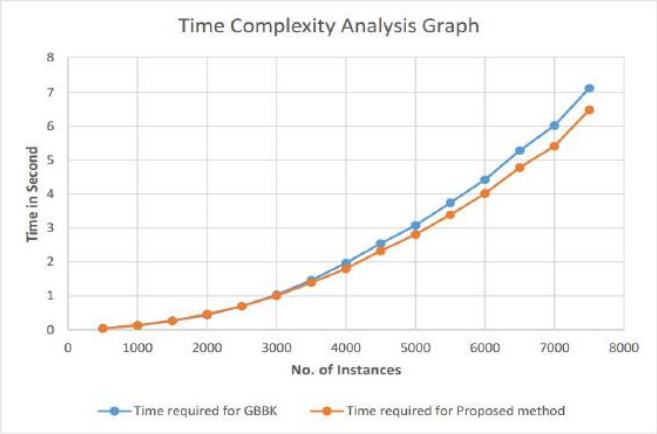

Figure-1: Comparison of the execution times for the proposed technique GBBK+ and GBBK

The Receiver Operating Characteristic is one of the metrics that is utilised in the process of establishing how effective the categorization system is. This statistic is known as the Receiver Efficiency (often abbreviated as ROC). Figure 2 is a depiction of the ROC that was obtained by applying SVM to the NSLKDD dataset. You can see this representation below. The True Positive Rate (TPR) is shown along the Y-axis of this graph, while the False Positive Rate (FPR) is displayed along the X-axis. These two rates are presented in the form of a percentage each. For the purpose of illustrating the ROC curve, a plot of the TPR vs the FPR may be used. After the elimination of the outlier, which made it possible to utilise the dataset for analysis, the ROC of the NSLKDD Train dataset can be seen in Figure 3, which shows the results of the study. The performance that was performed by the figure that came before this figure was somewhat more impressive than the performance that was produced by this figure.

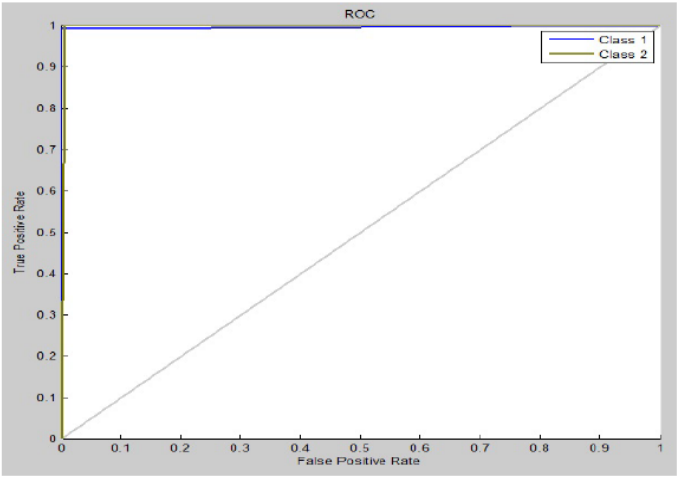

Figure-2: Prior to outlier elimination, the ROC

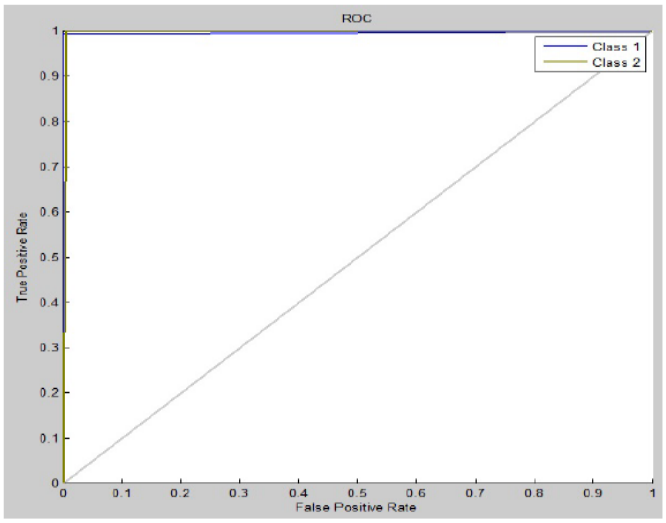

Figure-3: ROC after removal of outlier

6. DETECTION BASED UNSUPERVISED

The dataset of network connections that does not include any outliers and is shown in Figure-4 is the one that is used as the input for this stage. At this stage of the procedure, the objective is to come up with a collection of clusters that are based on the similarity measure. This phase will result in k+1 clusters, with the k+1st cluster comprising the objects that have not yet been assigned a classification and the other k clusters having the identical qualities in every instance. The unsupervised classification method k-point, which was published in [3,] carried out the needless comparison of objects in an iterative manner by decreasing the amount of characteristics with each passing iteration until it reached the threshold. [Citation needed] [Citation needed] [Citation needed] [Citation needed] [Citation needed] [Citation needed] [Citation needed] [(minimum attribute). Another disadvantage of this method is that the typical data label is assigned to the cluster that contains the most information, even if this is not always the case. For instance, the larger cluster that is used in a DOS assault is seen as belonging to the malevolent class rather than the ordinary class. This is because a DOS assault is an aggressive kind of attack.

The recommended approach, which has been given the term k- point+, has therefore been changed to handle these two limits as a consequence of this situation. The proposed method begins with unlabeled data and develops a clustering list based on the underlying statistical features of the data. This list is then used to cluster the data. These k random items were chosen from the dataset, and the other objects were clustered around them using the find sim similarity function (). This method will ultimately produce k plus one clusters from the whole unlabeled dataset that you provide.

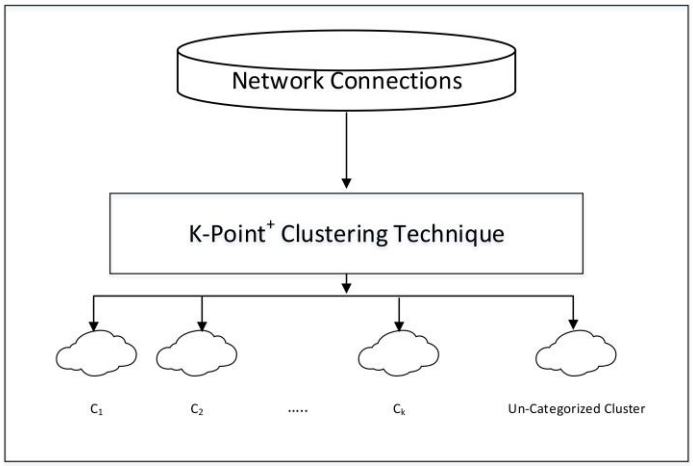

Figure-4: Unsupervised Classification Process

7. DETECTION UNDER SUPERVISION

In the process of supervised classification, a classification model known as the Support Vector Machine (more often abbreviated as SVM) is used. Supervised classification. This model is trained by using a training dataset that includes labels in order to make predictions. When an unsupervised classifier is applied to a set of data, the result is a list consisting of k+1 clusters. Following that, the aforementioned list is fed into a supervised classifier in order to be utilised as an input. To accurately tag the first k clusters on this list, you need to correctly estimate only one random object from throughout the full cluster. If it can be shown that this seemingly innocuous piece constitutes an attack, then the whole cluster will be labelled as an assault. If, on the other hand, it is not found out that this arbitrary item is part of an assault, then the cluster is recognised as being typical, as shown in Figure 5. However, the k+1th cluster comprises items that have not been categorised; hence, it is important to evaluate each object in this cluster in order to select the label for each individual object on an individual basis. Because of this, the total number of objects that are included into the model is equivalent to k plus the total number of objects that are contained inside the cluster that is denoted by k plus one. Once the label of the first item's k predicted label has been assigned as the label of the k cluster, the label of the cluster that each individual object was initially a member of is then assigned as the class label of each individual object. This is done after the label of the first item's k predicted label has been assigned as the label of the k cluster. Following the application of labels to every one of the products, the next step is to classify them as either safe to use or potentially dangerous. The intrusion detection system will not interfere with the regular connections; nevertheless, it will either prevent malicious connections from being made or notify the administrator to their presence if they are discovered. It is possible to dissect the operation of a support vector machine into two distinct components, both of which are shown in Figure 6. The first

phase is referred to as the online phase, while the second phase is referred to as the offline phase.

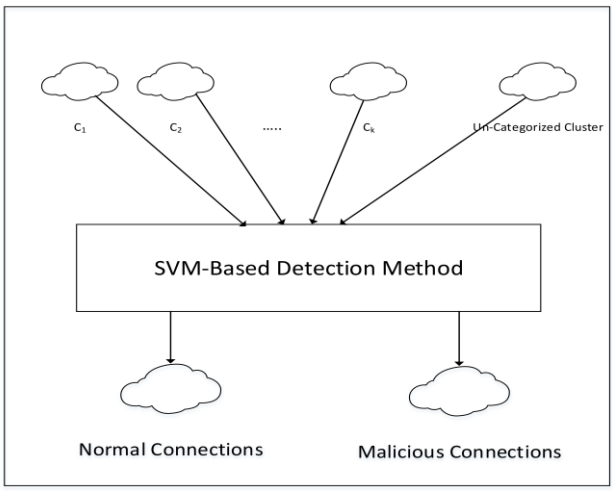

Figure-5: Supervised Classification Process

The phase of testing, which occurs at the same time as the period in which the game is being played online. During this stage of the process, we will produce our forecasts by using the model that we built during the phase before, which took place outside of the computer. The k-point+ methodology produce a collection of clusters that are made up entirely of feature sets as its end product. This output is then sent to a support vector machine in order for it to do an analysis on it. After that, the data are run through a support vector machine so that a prediction may be made about the class label of the feature sets. We made use of a binary classifier SVM model, which splits the whole output into two unique classes: the normal class, which is comprised of usual network connections, and the authorised category. We used this model to classify the data. This is the model that we based our work off of. The second category is known as the "malevolent class," and it contains all potentially harmful network connections. This category is referred to as "bad actors." These connections are either prohibited or a notification is sent to an administrator about them.

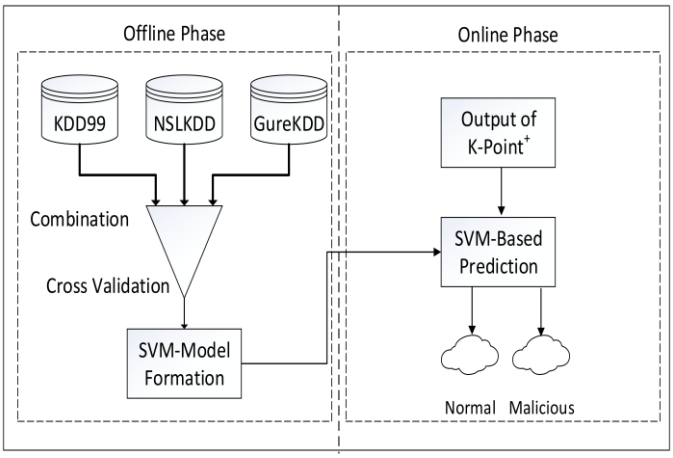

Figure-6: SVM Model's offline and online phases

8. CONCLUSIONS

This model illustrates a multi-stage intrusion detection system that is dependent on outlier, unsupervised, and supervised detection methodologies to identify potential threats. These techniques of detection are included in the model that has been supplied for your perusal. The GBBK+ algorithm produces results that are superior to those provided by the GBBK technique in terms of both the amount of time that is required and the precision of the detection. This is the case whether we are talking about the accuracy of the detection or the amount of time. When you look at the two algorithms side by side, you will see that this is the case. After the outliers in the dataset have been located with the assistance of the outlier detection method, the outliers themselves are removed from the compilation of the data. Due to the fact that the previous technique was able to successfully eliminate all outliers from the dataset, the performance of the second level was significantly improved as a direct consequence of this. Utilizing the method of finding data outliers is one way to successfully handle high-dimensional data in the shortest amount of time possible during processing. This may be accomplished by using the technique of recognising data outliers.

When the k-point method was first developed, there were a number of restrictions and drawbacks that needed to be solved before it could be considered a viable option. The k-point+ strategy that we developed allowed us to successfully complete this task. The empirical investigation came to the conclusion that the performance of k-point+ is superior than that of k-point in terms of the degree of difficulty of the task, the amount of time that is required, and the accuracy of the detection. The model that has been suggested is capable of detecting attacks with low frequency as well as high frequency, as well as assaults that are known and assaults that are unknown. Additionally, the model can identify assaults that are known and assaults that are unknown. In addition to that, the model is able to differentiate between attacks that are known and those that are unknown.

REFERENCE

1. Rebecca Bace and Peter Mell. Nist special publication on intrusion detection systems. Technical report, DTIC Document, 2001.

2. Behrouz A. Forouzan. Introduction to Cryptography and Network Security. McGraw-Hill Higher Education, 2008.

3. Prasanta Gogoi, DK Bhattacharyya, Bhogeswar Borah, and Jugal K Kalita. Mlh-ids: a multi-level hybrid intrusion detection method. The Computer Journal, 57(4):602{623, 2014.

4. Richard P Lippmann, David J Fried, Isaac Graf, Joshua W Haines, Kristopher R Kendall, David McClung, Dan Weber, Seth E Webster, Dan Wyschogrod, Robert K Cunningham, et al. Evaluating intrusion detection systems: The 1998 darpa line intrusion detection evaluation. In DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings, volume 2, pages 12{26. IEEE, 2000.

5. Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In Proceedings of the International Conference on Very Large Data Bases, pages 392{403. Citeseer, 1998.

6. Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. E_cient algorithms for mining outliers from large data sets. In ACM SIGMOD Record, volume 29, pages 427{438. ACM, 2000.

7. Wen Jin, Anthony KH Tung, Jiawei Han, and Wei Wang. Ranking outliers using symmetric neighborhood relationship. In Advances in Knowledge Discovery and Data Mining, pages 577{593. Springer, 2006.

8. Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and J□org Sander. Lof: identifying density-based local outliers. In ACM sigmod record, volume 29, pages 93{104. ACM, 2000.

9. Pedro Casas, Johan Mazel, and Philippe Owezarski. Unada: Unsupervised network anomaly detection using sub-space outliers ranking. In Jordi Domingo-Pascual, Pietro Manzoni, Sergio Palazzo, Ana Pont, and Caterina Scoglio, editors, NETWORKING 2011, volume 6640 of Lecture Notes in Computer Science, pages 40{51. Springer Berlin Heidelberg, 2011.

10. Charu Aggarwal and S Yu. An e_ective and efficient algorithm for high-dimensional outlier detection. The VLDB JournalThe International Journal on Very Large Data Bases, 14(2):211{221, 2005.

11. Graham Williams, Rohan Baxter, Hongxing He, Simon Hawkins, and Lifang Gu. A comparative study of rnn for outlier detection in data mining. In 2013 IEEE 13th International Conference on Data Mining, pages 709{709. IEEE Computer Society, 2002.

12. Neminath Hubballi, BidyutKr. Patra, and Sukumar Nandi. Ndot: Nearest neighbor distance based outlier detection technique. In Pattern Recognition and Machine Intelligence, volume 6744 of Lecture Notes in Computer Science, pages 36{42. Springer Berlin Heidelberg, 2011.

13. Zuriana Abu Bakar, Rosmayati Mohemad, Akbar Ahmad, and Mustafa Mat Deris. A comparative study for outlier detection techniques in data mining. In Cybernetics and Intelligent Systems, 2006 IEEE Conference on, pages 1{6. IEEE, 2006.

14. Kingsly Leung and Christopher Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38, pages 333{342. Australian Computer Society, Inc., 2005.

15. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an e_cient data clustering method for very large databases. In ACM SIGMOD Record, volume 25, pages 103{114. ACM, 1996.

16. Kalle Burbeck and Simin Nadjm-Tehrani. Adwice anomaly detection with real-time incremental clustering. In Information Security and Cryptology ICISC 2004, volume 3506 of Lecture Notes in Computer Science, pages 407{424. Springer Berlin Heidelberg, 2005.

17. Daniel Barbar_a, Julia Couto, Sushil Jajodia, and NingningWu. Adam: a testbed for exploring the use of data mining in intrusion detection. ACM Sigmod Record, 30(4):15{24, 2001.

18. M Ali Ayd_n, A Halim Zaim, and K G□okhan Ceylan. A hybrid intrusion detection system design for computer network security. Computers & Electrical Engineering, 35(3):517{526, 2009.

19. Jiong Zhang and Mohammad Zulkernine. A hybrid network intrusion detection technique using random forests. In Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on, pages 8{pp. IEEE, 2006.

20. Kai Hwang, Min Cai, Ying Chen, and Min Qin. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. Dependable and Secure Computing, IEEE Transactions on, 4(1):41{55, 2007.

21. Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. Expert systems with Applications, 29(4):713{722, 2005.

22. Zhi-Song Pan, Song-Can Chen, Gen-Bao Hu, and Dao-Qiang Zhang. Hybrid neural network and c4. 5 for misuse detection. In Machine Learning and Cybernetics, 2003 International Conference on, volume 4, pages 2463{2467. IEEE, 2003.