Indonesian Journal of Electrical Engineering and Computer Science
Vol. 23, No. 2, August 2021, pp. 953~961
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v23.i2.pp953-961

Intrusion detection system based on machine learning techniques

Musaab Riyadh, Dina Riadh Alshibani
Departement Computer Science, College of Science, Mustansiriyah University, Iraq

Article Info

Article history:

Received Oct 8, 2020
Revised Jun 29, 2021
Accepted Jul 7, 2021

Keywords:

Cluster feature
Incomplete data
Intrusion detection system
KNN
Loose cluster

ABSTRACT

Recently, the data flow over the internet has exponentially increased due to the massive growth of computer networks connected to it. Some of these data can be classified as a malicious activity which cannot be captured by firewalls and anti-malwares. Due to this, the intrusion detection systems are urgent need in order to recognize malicious activity to keep data integrity and availability. In this study, an intrusion detection system based on cluster feature concepts and KNN classifier has been suggested to handle the various challenges issues in data such as incomplete data, mixed-type, and noise data. To strengthen the proposed system a special kind of patterns similarity measures are supported to deal with these types of challenges. The experimental results show that the classification accuracy of the suggested system is better than K-nearest neighbor (KNN) and support vector machine classifiers when processing incomplete data set, inspite of droping down the overall detection accuracy.

Corresponding Author:

Musaab Riyadh
Department of Computer Science
Mustansiriyah University
Palestine street, Baghdad, Iraq
Email: m.shaibani@uomustansiriyah.edu.iq

1. INTRODUCTION

Cyber attacks have exponentially increased over the past decade; these attacks aim to steal the intellectual property of organization and disrupt their resoures and infra-structure [1]-[3]. Some of these attacks are insidious and cannot be detected by firewalls and antimalwares. Therefore, an additional security defensive line such as an Intrusion detection systems IDS are required to effectively monitor the activities of the network inorder to capture insidious attacks [4]. The intrusion detection systems IDS can be classified into two main approaches: signature-based (SIDS) and anomaly-based (AIDS) approachs. The main concept of SIDS is to compare the signature of current activity against a list of previously stored intrusions signatures and the alarm is triggered if a match is found. Due to this, the SIDS approach is hardly detecting a new attack which has no previous pattern in the database that represents the main weak point of this approach [5]. In the AIDS which is the focusing of this work, a model for the normal behavior of a computer system is build based on machine learning techniques, any remarkable deviation between the model and the observed behavior can be considered as an intrusion [6]. In contrary with SIDS approach, the update on data is not required to detect new attacks. Many researchers suggested AIDS based on single machine learing techniques such as support vector machine (SVM) [7], [8], the KNN algorithm [9], [10], and decision trees [11], [12]. The SVM and KNN classifiers are poorly performed with noisy and big data, while decision tree is a time-consuming classifier especially in training phase. The Bayesian Naïve is also suggested in [13], however this probabilistic classifier is not convenient for real time data that are generated with high speed. Other researchers proposed IDS based on hybrid techniques such as Zamani and Movahedi [14] suggested an accurate hybrid technique based on the gaussian mixture model (GMM) and K-means clustering algorithm and random forest classification technique. Saleh *et al.* [15] proposed a hybrid intrusion detection system depended on prioritized K-nearest neighbors and optimized support vector machine SVM classifiers but this system is not convenient for massive data with high dimensions. A hybrid real time IDS in [16] was proposed depending on two neural networks layers, the first neural network performs as an outliers-based detection for anonymous attacks and the others performs as a misuse-based detection for anonymous attacks. A more complex multi-level IDS was proposed by Al-Yaseen *et al.* [17] based on SVM and extreme learning machine. This system significantly enhanced the detection accuracy for different kind of attacks; however, the system was built for specific data set (KDD-Cup 99) and it is difficult to apply it to another data set. It is obvious that the hybrid techniques are more accurate than the single one but they are time consuming techniques. The aforementioned studues have focused only on enhancing the classification accuracy of the intrusion detection system and did not take into consideration the challenging issues in data set such as noisy and incomplete data. Besides that, they used iterative and complectated training techniqes which made it unsuitable for massive and incremental data.

On the other hand, IDS datasets have various challenges such as mixed-type, high dimensionality, and noisy data that significantly affect the classification accuracy. These challenges must be taken into consideration when designing efficient IDS [18], [19]. Various studies have been conducted to tackle these challenges: the studies in [20], [21] transformed n dimensional data of mixed-type to one dimensional data and classified these data based on KNN and SVM classifiers in order to maximize the efficiency of IDS. Manjunatha and Gogoi [22] proposed an efficient algorithm based on enhancing the Canberra method and minimum threshold support count to detect intrusions in high-dimensionality data set that consists of numerical and categorical features. Other studies have focused on the effects of noise in the performance of IDS. The works in [18], [23] eliminate the noisy patterns based on the density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm in order to enhance the classification accuracy of IDS. Bhosale *et al.* [24] suggested a noise removal algorithm to enhance the classification accuracy of Naive Bayes classifier however, it is a time-consuming classifier. Hussain and Lalmuanawma [25] proved that self organization map has better intrusion detection accuracy in noise data than widespread classifiers (JRip, J48, RF, NBTree) despite of the low performance in normal data. These studies focused on the importance of eliminating noise to enhance the classification accuracy. However, theses studies supported similarity measures such as Euclidean distance which are significantly affected when using incomplete data. Table 1 shows a compartion between the related works. Ultimately, an intrusion detection has been proposed in this study to handle various challenging issues in massive data sets such as mixed-type, high dimensionality, noisy, and incomplete data. To the best of our knowledge there are no studies that focus on the the problem of incomplete data set due to intentional or unintended errors in collecting data which is the main objective of this study.

Table 1. Related works comparsion

Work	Techniques	Data Set	Mixed type data	Noisy data	Incomplete data	Evaluation method
Saleh *et al*. [15]	GMM and K-means	KDD-Cup99	Yes	No	No	AC[1], FAR[2], DR[3]
Al-Yaseen *et al*. [17]	SVM and extreme learning machine	Only KDD-Cup99	Yes	No	No	DR, AC, FAR
Dong *et al*. [18]	K-means + DBSCAN	NSL-KDD	Yes	Yes	No	AC, Precision
Chen *et al*. [19]	DBSCAN	DARPA	Yes	Yes	No	TDR[4], FDR[5]
Guo *et al*. [20]	SVM	KDD-Cup99				DR, ROC[6]
Lin *et al*. [21]	K-means + KNN	KDD-Cup99	Yes	No	No	AC
Manjunatha *et al*. [22]	Canberra method and MTSC[7]	KDD-Cup99	Yes	No	No	AC
Shakya *et al*. [23]	K-means + DBSCAN + SMO[8]	KDD-Cup99	Yes	Yes	No	AC
Bhosale *et al*. [24]	Naive Bayes	KDD-Cup99	Yes	Yes	No	AC, Precision
Hussain *et al*. [25]	NN(SOM[9])	KDD-Cup99& NSL-KDD3	Yes	Yes	No	AC, TPR[10], FPR[11], ROC

[1]Accuracy [2]False Alarm Rate [3]Detection Rate [4]True Detection Rate [5]False Detection Rate [6]Receiver Operating Characteristic
[7]Minimum Threshold Support Count [8]Sequential Minimal Optimization [9] Self-Organizing Map [10]True Positive Rate [11]False Positive Rate

2. DISSIMILARITY MEASURE

The distance (dissimilarity) between a pair of patterns is an essential task to evaluate how alike or unalike patterns are in comparison to one another. It is the essence of different machine learning applicaions such as clustering and classification which remarkably affects the classification accuracy [26], [27]. Most of the existing studies support euclidean distance (ED) to measure the dissimilarity between two patterns of mixing attributes (e.g. binary; nominal; ordinal; and numeric), however ED is sensitive to incomplete data. Therefore, a special kind of dissimilarty measure has been employed in this study to process the mixed-type attributes that have missing values for some attributes [28] as defined in (1).

$$Dist(p_i, p_j) = \frac{\sum_{A=1}^{N} \mu_{p_i p_j}^{A} dist_{p_i p_j}^{A}}{\sum_{A=1}^{N} \mu_{p_i p_j}^{A}} \quad (1)$$

Where dist (p_i,p_j) is the dismilarity measure between patterns p_i, p_j and *N* represents the number of attributes in each pattern, and the parameter $\mu_{p_i p_j}^{A}$ =0 either:

- If there is no measurments of attribute A of patterns p_i or p_j.
- If A is asymmetric binary attribute and p_i^A=0, p_j^A =0.
- Otherwise, $\mu_{p_i p_j}^{A}$ =1.

The contribution of attribute A to the distance (dissimilarity) between p_i and p_j is calculated based on its type:

- If attribute A is a numeric type: $dist^A_{pi,\, pj} = |\, x_{pi}^A - x_{pj}^A\, |/(Max^A - Min^A)$, where max ^{A} and min ^{A} are the maximum and minimum values of the attribute A over all the none missing values.
- If attribute A is a nominal type or binary: $dist^A_{pi,\, pj}$ =0 if $p_i^A = p_j^A$; otherwise, $dist^A_{pi,\, pj}$ =1.
- If attribute A is ordinal type: convert the rank of attributes r_{pi}^A and r_{pj}^A to z_{pi}^A and z_{pj}^A as given in (2).

$$z_p^A = (r_p^A - 1)/(M_A - 1) \quad (2)$$

Where M_A is the possible states number that an ordinal attribute can have. Then compute the dissimilarity as defined in (3):

$$dist^A_{pi,\, pj} = |z_{pi}^A - z_{pj}^A| \quad (3)$$

Finally, the supported similarity measure combines the various attributes into a single dissimilarity measure onto a common scale of the interval [0.0, 1.0].

3. THE RESEARCH METHOD

The main objective of this study is to design an intrusion detection system for the incomplete data (IDS-ID) classifier based on hybrid machine learning techniques that are capable to deal with incomplete data set along with the other challenges such as mixed-type and noise data set. The proposed classifier IDS-ID consists of two phases: the training phase and the testing phase. The training phase aims to cluster the data based on the notion of cluster features CFs, when the entire size of CFs exceeds a given memory space the most similar CFs is merged. While, the KNN classifier has been supported in the testing phase. Finally, 50% of the data set is used for training phase and 50% for testing phase as illustrated in Figure 1.

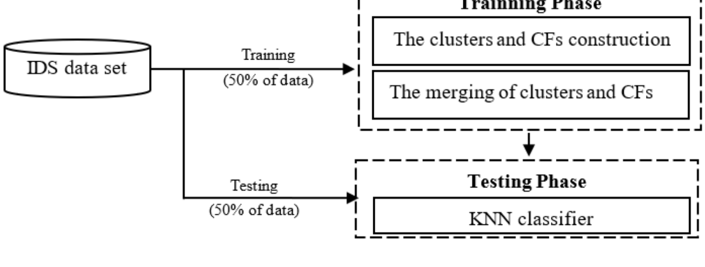


Figure 1. The IDS-ID phases

3.1. The training phase

The training phase of this study is mainly based on the notion of cluster features CF due to its good speed and scalability in massive or even streaming databases. It consists of two levels: i) the construction of clusters and CFs and ii) the merging of clusters and CFs. The first level stores summarizing information about each cluster in CF data structure and update this information once a new pattern is added to the cluster e.g the CF_i of cluster C_i is (N, Ls_1, Ss_1, Ls_2, Ss_2, …., Ls_m, Ss_m) where *N* represents the number of patterns in the cluster, *m* is the number of features in each pattern and Ls_m, Ss_m represent the linear sum and square sum of feature *m* for all patterns in C_i. At the end of the construction level, the clusters that have patterns less than $Item_{min}$ threshold will be discard immediately since they are noise data. In the merging level, the most similar clusters are merged as defined in (4). The merging task is based on a dissimilarity measure that find a typical trade-off between clusters density and the distance between their centers as defined in (5). Note that, the merging level is activated when the last pattern in the data set is processed.

$$\text{Merge}\ (CF_i, CF_j) = (N_i+N_j, LS_i^1+LS_j^1, SSi^2+SS_j^2\ \ldots., LS_i^m+LS_j^m, SS_i^m+SS_j^m) \tag{4}$$

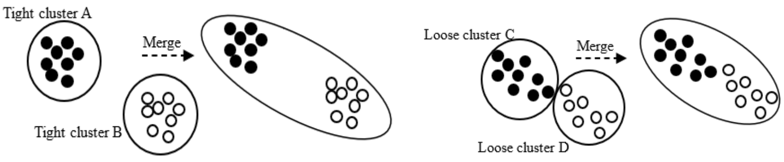
$$\text{Distance}\ (C_i, C_j) = |C_{ceni}-C_{cenj}|-0.5(C_{Di}+ C_{Dj}) \tag{5}$$

Where C_{ceni}, C_{cenj} are the center of clusters C_i and C_j, and C_{Di}, C_{Dj} represent the clusters density and can compute from CF_i, CF_j parameters based on (6) and (7).

$$C_{cen} = LS/N \tag{6}$$

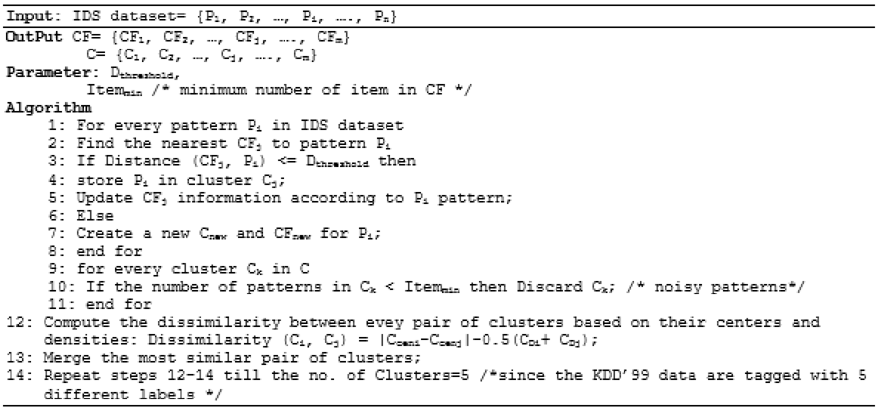
$$C_D = \sqrt{\frac{2*N*SS-LS^2}{N*(N-1)}} \tag{7}$$

It is obvious that (5) gives a priority to merging two loose clusters together rather than merging tight clusters if the distance between their centers is approximately equals. This is because, the merging two tight clusters will break their tightness as illustrated in Figures 2(a) and (b). The clusters' merging process is continued till the number of clusters in the training phase becomes equal to five. This due to, the KDD-Cup99 data are tagged with 5 different labels. The main steps of training phase are illustrated in Figure 3. Finally, The CFs technique has been chosen for this level due to their ability to cluster high dimensional data with a single pass over the data which lead to significantly minimize the running time of the training phase.



(a) (b)

Figure 2. Merging clusters; (a) merging tight cluster and (b) merging loose cluster



```
Input: IDS dataset= {$P_1$, $P_2$, …, $P_i$, …., $P_n$}
OutPut CF= {$CF_1$, $CF_2$, …, $CF_j$, …., $CF_m$}
        C= {$C_1$, $C_2$, …, $C_j$, …., $C_m$}
Parameter: $D_{threshold}$,
        $Item_{min}$ /* minimum number of item in CF */
Algorithm
     1: For every pattern $P_i$ in IDS dataset
     2: Find the nearest $CF_j$ to pattern $P_i$
     3: If Distance ($CF_j$, $P_i$) <= $D_{threshold}$ then
     4: store $P_i$ in cluster $C_j$;
     5: Update $CF_j$ information according to $P_i$ pattern;
     6: Else
     7: Create a new $C_{new}$ and $CF_{new}$ for $P_i$;
     8: end for
     9: for every cluster $C_k$ in C
     10: If the number of patterns in $C_k$ < $Item_{min}$ then Discard $C_k$; /* noisy patterns*/
     11: end for
12: Compute the dissimilarity between evey pair of clusters based on their centers and
    densities: Dissimilarity ($C_i$, $C_j$) = $|C_{ceni}-C_{cenj}|-0.5(C_{Di}+ C_{Dj})$;
13: Merge the most similar pair of clusters;
14: Repeat steps 12-14 till the no. of Clusters=5 /*since the KDD'99 data are tagged with 5
    different labels */
```

Figure 3. The training phase of IDS-ID classifier

3.2. The testing phase

The second phase of the IDS-ID system is the testing phase, it aims to classify the testing data based on the k-nearest neighbors (KNN) algorithm. The KNN classifier has been utilized in this phase due to its low computation cost, few parameters, and classify the data based on non-linear decision boundaries. The key notion behind the KNN classifier is that, for every pattern in the training set, if most of its K nearest neighbor patterns belong to a cluster C_i, then the pattern belongs to the C_i cluster [9]. The KNN classifier in this phase has used 50% of the KDD-Cup99 to test the classification accuracy of the IDS-ID classifier.

4. RESULTS AND DISCUSSION

The KDD-Cup99 data set is used in this work to train and test the classifiers IDS-ID, KNN, and SVM in order to evaluate their performance, it was created and managed by DARPA based on the orginal data set from MIT Lincoln Laboratories to evaluate the researchs on an intrusion detection systems. The data set contains 494,020 records and each record has 41 dimensions of various types (binary, nominal, and numeric). These dimensions represent the network conections and can be categorized into three main classes: the traffic flow (19 dimensions), the intrinsic types (9 dimensions), and the content type (13 dimensions). Table 2 illustrates a small sample of these dimensions. The IDS-ID classifer has been implemented using Matlab (2012a) on intel core i3 hp laptop with windows 7 oprating system.

Table 2. TCP conection attributes in KDD-cup99 data set

Category Name	Features Names	Data type	Description
Basic Attributes	Duration	Integer	connection time (seconds)
	Protocol type	Nominal	Protocol kind (TCP and UDP)
Content Attributes	Logged in	Binary	1 if success to login otherwizeb 0.
	Number of unsuccessful Logins	Interger	Number of unsuccessful logins into a connection
Traffic Attributes (2s time window from dest. to host)	destination host count	Integer	connections Sum to the same destination IP address
	destination host same srv rate	Real	Percentage of connections that were to the same service, among the connections aggregated into destination host count (P32)

Note that, the KDD-Cup99 patterns are taged with five different labels: the first one represents the authorized activites and the other four labels represent four kinds of attack activities as listed below:

- Denial-of-service (Dos): some computer resources are unavailable or too busy to response to legitimate user's requests.
- Remote-to-local (R2l): unauthorized access is done by a remote device in order to detect the vulnerabilities of the machine.
- User-to-root U2r: an unauthorized attack trys to access the privileges of local superuser (root) through the system vulnerabilities.
- Probe Prb: an unauthorized attack attempt to get information about the target host to find vulnerabilities.

4.1. Data pre-processing

Pre-processing is a crucial step involved while dealing with data sets before using it for data analysis and construction classifier models. Various pre-processing steps have been done in this study such as: firstly, eliminating duplicates from the dataset. Secondly, mapping some nominal attributes to numeric-valued. Thirdly, mapping some nominal attributes (e.g. "protocol" and “TCP status flag”) to binary attributes. Finally, normalizing some attributes since they have different scales such as "destination host count” which has a range of (0-255), whereas “sourcebytes” ranges (from 0 to 693375630).

4.2. Parameters sensitivity

The parameters $D_{threshold}$ and *K* related to KNN classifier in the testing phase are significantly affecting the classification accuracy of the IDS-ID classifier. Therefore, the classification accuracy of IDS-ID has been tested for different values of $D_{threshold}$ (from 2 to 8) and K (3, 5, 7, 9, and 11) to determine the values which give the higher classification accuracy for the IDS-ID classifier. The experiments illustrate that the best values that perform the best classification accuracy (98.49) are when $D_{threshold}$ =4 and k=5 as illustrated in Table 3.

Table 3. The accuracy detection for different values of $D_{threshold}$ and k based on selected 20 dimensions

$D_{threshold}$	K=3	K=5	K=7	K=9	K=11
2	97.6	97.49	96.5	95.56	97.8
4	96.78	98.49	96.4	95.40	96.70
6	95.70	97.90	96.10	95.35	96.25
8	95.60	97.65	95.91	94,90	96.14

4.3. Efficiency evaluation

The efficiency (running time) of IDS-ID classifier has been compared with KNN and SVM classifiers based on 20 selected dimensions as illustrated in Table 4. The comparison shows that the running time of data pre-processing phase for IDS-ID took longer time than KNN. However, the overall running time for IDS-ID (1504 mins) is less than KNN (2173 mins) and SVM (4155 mins) classifiers since the IDS_ID classifier is based on CF concepts which significantly minimize the running time.

Table 4. The running time of the IDS-ID, SVM, and KNN based on selected 20 dimensions

		Data preprocessing	Training and testing	overall
KDD-Cup99 data set	KNN	20 mins	2153 mins	2173 mins
	SVM	-	4155 mins	4155 mins
	IDS-ID	30 mins	1504 mins	1504 mins

4.4. The classification performance

In this section, the classification accuracy of the IDS-ID classifier is compared with the performance of KNN, and SVM classifiers based on the detection rate (DR), false positive rate (FR), and accuracy (A) [3]. These metices are used by the most existing studies and defined in (8)-(10).

$$DR = TP/(TP + FN) \tag{8}$$

$$FR = FP/(FP + TN) \tag{9}$$

$$A = (TN + TP)/(TN + TP + FP + FN) \tag{10}$$

Where:
False positive (FP) is the normal patterns number, which is classified as an attack instances.
False negative (FN) is the attacks patterns number, which are classified as a normal instances.
True positive (TP) is the detected attacks number and in fact they are attacks.
True negative (TN) is the detected normal instances number and in fact they are normal.

The first step to evaluate the performance of the IDS-ID classifier is to find the confusion matrix based on the KDD-Cup99 as elaborate in Table 5. It is obvious that (98.49%) of the normal patterns can be classified correctly, while the performance of IDS-ID shows a low classification rate toward U2r (5.4%) and R2l (6.46%) attacks. In addition, four experiments have been done to assess the performance of the IDS-ID, KNN, and SVM classifiers based on KDD-Cup99 data set: the first experiments used the actual data without any change. The final results shows that the overall accuracy of IDS-ID (92.85) is better than KNN (91.53), and SVM (92.25) as illustrated in Table 6. However, the diffirence between the classification accuracy of the three classifiers is small.

Table 5. Confusion matrix obtained with IDS-ID for the five classes of the KDD-cup99

	Normal	Prb	R2l	Dos	U2r	Actual	Correct
Normal	59587	598	80	145	88	60498	98.49%
Prb	415	3640	21	79	5	4160	87.5%
R2l	15003	81	1055	23	52	16214	6.46%
Dos	4350	1525	517	223460	0	229852	97.21%
U2r	55	158	12	0	13	238	5.4%

Table 6. Classification accuracy of KNN, SVM, and IDS-ID for the KDD-cup99

	Metric	KNN	SVM	IDS-ID
Normal	DR%	97.26	96.45	98.49
	FR%	8.85	8.65	8.55
Prb	DR%	80.45	85.35	86.25
	FR%	0.40	0.40	0.75
R2l	DR%	6.45	7.20	6.20
	FR%	0.1	0.2	0.21
Dos	DR%	97.15	96.95	97.55
	FR%	0.55	0.95	0.40
U2r	DR%	11.8	9.20	4.36
	FR%	0	0.1	0
Overall	AC%	91.53	92.25	92.85

In the second experiment, the overall classification accuracy of the all classifiers have been slightly decreased due to removing 5% of the data but still the the IDS-ID classifier (92.24) has the highest classification accuracy. Besides that, the classification accuracy gap between the IDS-ID classifier and, KNN and SVM classifiers has been increased as illustrated in Table 7. The classification accuracy is still decreasing in the third and fourth experiments of the all classifiers due to increasing the percentage rate of the removing data 10 and 15%. However, the classification performance of the IDS-ID classifier (91.0) is still the highest as illustrated in Tables 8 and 9. Ultimately, the classification accuracy of IDS-ID classifier is better than KNN and SVM when randomly removing 5, 10, and 15 % of the data inspite of droping down the overall detection accuracy of all classifiers as shown in Figure 4.

Table 7. Classification accuracy of KNN, SVM, and IDS-ID after randomly eliminate 5% of the KDD-cup99 data set

	Metric	KNN	SVM	IDS-ID
Normal	DER%	96.1	95.9	97.9
	FPR%	8.5	8.24	8.15
Prb	DER%	78.2	84.5	86.7
	FPR%	0.40	0.41	0.73
R2l	DER%	5.73	6.21	5.9
	FPR%	0.08	0.17	0.18
Dos	DER%	96.10	96.00	97.00
	FPR%	0.41	0.81	0.34
U2r	DER%	11.3	8.95	4.52
	FPR%	0	0.09	0
Overall	AC%	88.94	89.34	92.24

Table 8. Classification accuracy of KNN, SVM, and IDS-ID after randomly eliminate 10% of the KDD-cup99 data set

	Metric	KNN	SVM	IDS-ID
Normal	DER%	96.85	96.35	97.15
	FPR%	8.56	8.31	8.26
Prb	DER%	78.2	84.55	86.6
	FPR%	0.40	0.42	0.70
R2l	DER%	5.75	6.24	5.99
	FPR%	0.08	0.17	0.18
Dos	DER%	96.2	96.05	97.01
	FPR%	0.39	0.79	0.32
U2r	DER%	11.10	8.80	4.28
	FPR%	0	0.07	0
Overall	AC%	88.35	88.88	91.0

Table 9. Classification accuracy of KNN, SVM, and IDS-ID after randomly eliminate 15% of the KDD-cup99 data set

	Metric	KNN	SVM	IDS-ID
Normal	DER%	96.55	96.05	97.01
	FPR%	8.50	8.24	8.20
Prb	DER%	77.9	84.05	86.1
	FPR%	0.38	0.41	0.67
R2l	DER%	5.65	6.11	5.78
	FPR%	0.07	0.16	0.17
Dos	DER%	95.9	95.8	96.89
	FPR%	0.37	0.77	0.30
U2r	DER%	10.91	8.62	4.07
	FPR%	0	0.05	0
Overall	AC%	87.6	88.33	89.7

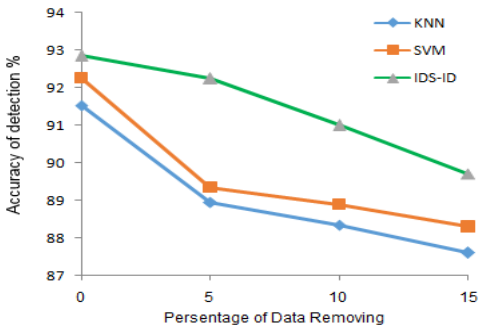


Figure 4. The dropping in detection accuracy after removing 5, 10, and 10 % of the data

5. CONCLUSION

The analysis of the intrusion detection data set based on machine learning techniques is a challenging task due to its massive size, mixed-type attributes, and the redundancy of data. Besides that, the data may be incomplete and noisey. In this study, an intrution detection system has been proposed to tackle these issues, it connsists of two phases: the learing phase and testing phase. The learning phase supports the cluster feature concept to summarize the data set and special kind of similarity measures to deal with mixed-type attributes and incomplete data. While the testing phase uses the KNN calssifer due to its low computational cost. The experimental results shows that the proposed classifier has a higher classification accuracy and lower running time in actual data and incomplete data when randomly remove 5, 10, and 15% persantege of data inspite of droping down the overall detection accuracy as compared with SVM and KNN classifier.

ACKNOWLEDGEMENTS

The author would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq Baghdad-Iraq) for it supporting the present work.

REFERENCES

[1] S. K. Sahu and D. P. Mohapatra, "A Review on Scalable Learning Approaches on Intrusion Detection Dataset," *Proceedings of ICRIC Springer*, vol. 597, pp. 699-714, 2020, doi: 10.1007/978-3-030-29407-6_50.

[2] M. Pradhan, C. K. Nayak, and S. K. Pradhan, "Intrusion Detection System (IDS) and Their Types," in *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications*, IGI Global, pp. 481-497, 2020, doi: 10.4018/978-1-5225-9866-4.ch026.

[3] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117-123, 2016, doi: 10.1016/j.procs.2016.06.016.

[4] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493-501, 2019, doi: 10.1007/s12083-017-0630-0.

[5] A. Khraisat, I. Gondal, P. Vampley, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1-20, 2019, doi: 10.1186/s42400-019-0038-7.

[6] N. Ugtakhbayar B. Usukhbayar, and S. Baigaltugs, "A Hybrid Model for Anomaly-Based Intrusion Detection System," *Proceedings of Advances in Intelligent Information Hiding and Multimedia Signal Processing Springer*, pp. 419-431, 2020, doi: 10.1007/978-981-13-9710-3_44.

[7] S. Krishnaveni, *et al*., "Anomaly-Based Intrusion Detection System Using Support Vector Machine," *Proceedings of Artificial Intelligence and Evolutionary Computations in Engineering Systems springer*, 2020, pp. 723-731.

[8] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Systems*, vol. 136, pp. 130-139, 2017, doi: 10.1016/j.knosys. 2017.09.014.

[9] Y. Liao, and V. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & security*, vol. 21, no. 5, pp. 439-448, October 2002, doi: 10.1016/S0167-4048(02)00514-X.

[10] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, no. 5, pp. 1-8, 2014, doi: 10.1155/2014/240217.

[11] Y. J. Chew, S. Y. Ooi, Kok-Seng Wong, and Y. H. Pang, "Decision Tree with Sensitive Pruning in Network-based Intrusion Detection System," *Proceedings of Computational Science and Technology Springer*, vol. 603, pp. 1-10, 2020, doi: 10.1007/978-981-15-0058-9_1.

[12] S. M. Mousavi, V. Majidnazhad, and A. Naghipour, "A new intelligent intrusion detector based on ensemble of decision trees," *Journal of Ambient Intelligence and Humanized Computing*, 2019, pp. 1-13, doi: 10.1007/s12652-019-01596-5.

[13] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, 2001, pp. 38-49, doi: 10.1109/SECPRI.2001.924286.

[14] M. Zamani and M. Movahedi, "Machine Learning Techniques for Intrusion Detection," *arXiv preprint arXiv:1312.2177*, 2013.

[15] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers," *Artificial Intelligence Review*, vol. 51, no. 3, pp. 403-443, 2019, doi: 10.1007/s10462-017-9567-1.

[16] G. Mylavarapu, J. Thomas, and A. K. TK, "Real-Time Hybrid Intrusion Detection System Using Apache Storm," *2015 IEEE 12th International Conference on Embedded Software and Systems*, 2015, pp. 1436-1441, doi: 10.1109/HPCC-CSS-ICESS.2015.241.

[17] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296-303, Jan. 2017, doi: 10.1016/j.eswa.2016.09.041.

[18] G. Dong, Y. Jin, S. Wang, W. Li, Z. Tao, and S. Guo, "DB-Kmeans:An Intrusion Detection Algorithm Based on DBSCAN and K-means," *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1-4, doi: 10.23919/APNOMS.2019.8892910.

[19] Z. Chen and Y. Li, "Anomaly detection based on enhanced DBScan algorithm," *Procedia Engineering*, vol. 15, pp. 178-182, 2011, doi: 10.1016/j.proeng.2011.08.036.

[20] C. Guo, Y. Zhou, Y. Ping, Z. Zhang, G. Liu, and Y. Yang, "A distance sum-based hybrid method for intrusion detection," *Applied intelligence*, vol. 40, no. 1, pp. 178-188, 2014, doi: 10.1007/s10489-013-0452-6.

[21] Wei-Chao Lin, Shih-Wen Ke, and Chih-Fong Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13-21, April 2015, doi: 10.1016/j.knosys.2015.01.009.

[22] B. Manjunatha and P. Gogoi. "Anomaly based intrusion detection in mixed attribute dataset using data mining methods," *Journal of Artificial Intelligence*, vol. 9, no. 1-3, pp. 1-11, 2016.

[23] V. Shakya and R. R. S. Makwana, "Feature selection based intrusion detection system using the combination of DBSCAN, K-Mean++ and SMO algorithms," *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 2017, pp. 928-932, doi: 10.1109/ICOEI.2017.8300843.

[24] K. S. Bhosale, M. Nenova, and G. Iliev, "Modified Naive Bayes Intrusion Detection System (MNBIDS)," *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 2018, pp. 291-296, doi: 10.1109/CTEMS.2018.8769248.

[25] J. Hussain and S. Lalmuanawma, "Feature analysis, evaluation and comparisons of classification algorithms based on noisy intrusion dataset," *Procedia Computer Science*, vol. 92, pp. 188-198, 2016, doi: 10.1016/j.procs.2016.07.345.

[26] X. Zuo, Z. Chen, L. Dong, J. Chang, and B. Hou, "Power information network intrusion detection based on data mining algorithm," *The Journal of Supercomputing*, vol. 76, no. 7, pp. 1-19, 2019, doi: 10.1007/s11227-019-02899-2.

[27] M. Riyadh, N. Mustapha, and D. Riyadh, "Review of Trajectories Similarity Measures in Mining Algorithms," *2018 Al-Mansour International Conference on New Trends in Computing, Communication, and Information Technology (NTCCIT)*, 2018, pp. 36-40, doi: 10.1109/NTCCIT.2018.8681186.

[28] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*, USA: Morgan Kaufmann, 2011.