Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection

Mahendra Prasad[a,*], Sachin Tripathi[a], Keshav Dahal[b]

[a]*Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, India*
[b]*School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley, UK*

Abstract

The massive growth of data in the network leads to attacks or intrusions. An intrusion detection system detects intrusions from high volume datasets but increases complexities. A network generates a large number of unlabeled data that is free from labeling costs. Unsupervised feature selection handles these data and reduces computational complexities. In this paper, we have proposed a clustering method based on unsupervised feature selection and cluster center initialization for intrusion detection. This method computes initial centers using sets of semi-identical instances, which indicate dense data space and avoid outliers as initial cluster centers. A spatial distance between data points and cluster centers create micro-clusters. Similar micro-clusters merge into a cluster that is an arbitrary shape. The proposed cluster center initialization based clustering method performs better than basic clustering, which takes fewer iterations to form final clusters and provides better accuracy. We simulated a wormhole attack and generated the Wormhole dataset in the mobile ad-hoc network in NS-3. This work has executed on different network datasets (KDD, CICIDS2017, and Wormhole dataset), which outperformed for new attacks or those attacks contain few samples. Experimental results confirm that the proposed method is suitable for LAN and mobile ad-hoc network, varying data density, and large datasets.

Keywords: Unsupervised intrusion detection, Unsupervised feature selection, Cluster center initialization, Clustering, Mobile ad-hoc network,

*Corresponding author
Email addresses: je.mahendra@gmail.com (Mahendra Prasad), var_1285@yahoo.com (Sachin Tripathi), Keshav.Dahal@uws.ac.uk (Keshav Dahal)

Wormhole attack.

1. Introduction

The exponential growth of network size and traffic carried out the attention of network security. An unauthorized user can misuse or harm network resources. However, a firewall as a first-line defense mechanism is used for intrusion detection. It is not powerful enough to detect and prevent intrusions. An attacker can faithfully bypass the first-line mechanism. Antivirus software is a second-line defense system, that can detect only stored signature of attack. An Intrusion Detection System (IDS) is a mechanism that detects malicious behavior, unauthorized access, and quickly prevents them from network communication. Moreover, IDS is a strong detection system as it gathers intrusion information or related behaviors that intrude security policies. Intrusion information is an important asset of computer network security hence the attacker tries to conceal his/her history and dynamic nature. There are mainly two categories of IDS misuse detection system and anomaly detection system. These defense mechanisms may combine and form a hybrid detection system. Misuse detection system detects only those signatures are present in the database, whereas the anomaly detection system computes deviation of behaviors from baseline profile [Mishra et al., 2018]. The main aim of unsupervised IDS to increase Detection Rate (DR) and reduce False Alarm Rate (FAR). The IDS detects an intrusion in communication, then generates an alarm and sends it to the network administrator to prevent intrusion. To maintain network security, the main problem of machine learning is computational complexity as dimension and size of the dataset get larger.

Feature selection approach reduces dimension and size by selecting a subset of essential features. It can be applied for both mode of training model as supervised and unsupervised by whether label of information is known or unknown [Xie et al., 2018]. We apply Unsupervised Feature Selection (UFS) [Zhu et al., 2017] when class labels of data are unavailable. The UFS solves the noticeable problems of clustering methods such as reduce the additional computational cost and increase performance of the system. Redundant or irrelevant features may lead to overfitting problem, which degrade the performance of the system. Here, the main objective of UFS method is to keep the training model as small as possible, that reduces insignificant features or those have a negligible effect [Wang et al., 2019]. The best subset of features can contain a minimal number of features and contribute system accuracy as much possible. This is a fundamental mechanism of processing the dataset and provide a suitable approach to dimensionality reduction. It can be employed for many applications such as text, images, videos, and genes analysis.

This method can extensively enhance the interpretability of machine learning algorithms, and an algorithm integrated with feature selection often present better generalization. The proposed UFS method finds a suitable subset of predictive features.

Many machine learning techniques are used for clustering, which indicates the significance of UFS. Clustering method partitions data into clusters in many iterations [Gong et al., 2018]. Some difficulty has been considerably reported in many research such as (1) the clustering method assumes that the number of clusters is known by users, which is not true in practice [Masud et al., 2018], (2) an iterative method is sensitive to initial cluster centers especially in the K-means algorithm, (3) K-means algorithm can be stuck on local minima. The main reason of such difficulties of iterative clustering is random center initialization, and these problems solve using Cluster Center Initialization (CCI) based clustering. A selection of initial cluster centers is essential that have major role in the formation of final clusters [Huang et al., 2017, Peng and Liu, 2018].

Naturally, clusters are in non-convex (arbitrary) shapes [Hyde et al., 2017], whereas distance based methods always find convex shapes of clusters. The proposed method is arbitrary shape clustering, that differs from density-based clustering. It identifies the number of micro-clusters and work for varying density data. We have executed the proposed method on an extremely imbalanced (KDD and CICIDS2017) datasets; whereas, the KDD test set includes many attacks as unknown attacks. The experimental results show the performance in increasing order as K-means, micro-clustering (New K-means), and CCI based micro-clustering (proposed). This work measures different characteristics (parameters) and executed on different network datasets that show the reliability of the system. The KDD dataset was generated through testbeds of the Air force environment Local Area Network (LAN), and the CICIDS2017 is recently generated dataset; whenever, wormhole dataset is simulated in Mobile Ad-hoc NETwork (MANET) environment. Moreover, the proposed method performs better than existing unsupervised intrusion detection mechanisms. The main contributions are enumerated below.

(i) Feature selection is an important technique to select the best subset of features and produce better results [Ambusaidi et al., 2016]. We have proposed an UFS method that select more informative features. It also reduces dimension and size of the dataset.

(ii) Basic clustering algorithm randomly selects initial centers and iteratively computes centroids [Ma et al., 2015]. We have proposed CCI based clustering algorithm that resolves notable problems. This method

computes semi-identical instances and gives many sets. A mean of data points present in the set as initial centroid. The advantages of initialization of cluster center over basic approach are (1) it avoids outliers as initial centers, (2) a new initial cluster center is mean of data points present in the set, that indicates dense data space, (3) it reduces iterations of clustering.

(iii) We have applied K-means clustering to group similar data [Zhao et al., 2018]. In proposed method, K is the number of micro-clusters which is greater than c. Finally, it merges micro-clusters into c-clusters. Experimental results show better detection rate of new attacks or those attacks have few samples. The CCI based clustering method provides better accuracy and execution time than existing clustering mechanisms.

The rest of paper is organized as follows : Section 2 reviews related literatures, and Section 3 introduces the importance of unsupervised intrusion detection system for mobile ad-hoc network. Section 4 provides detail analysis of KDD, CICIDS2017, and Wormhole dataset. The successive Section 5 describes the proposed method. Experiments are conducted and the performance of the proposed method is examined in Section 6. Finally, we conclude the paper with future direction in Section 7.

2. Related work

High dimensional dataset contains more information but also introduce redundancies and noises, that increases the complexity of clustering algorithms. There are many feature selection methods, but most of them under the supervised mode of training. Unsupervised feature selection is a difficult task due to unknown labels. Recently, some works have shown the significance of UFS, such as score of features [Wang et al., 2019], feature similarity, clustering based non-negative matrix factorization, clustering based co-regularization method, locality of data, maximization of distance of different clusters, relevance feature representation with spatial geometrical structure of data, joint sparse matrix, non-negative spectral analysis, low rank structure preserving, locality structure preserving, matrix factorization, inner product regularization, self expressing model, and relationship among features. The proposed UFS method is based on the occurrence of objects, and it easily computes features score.

It is extremely important to adopt the best initial cluster center in the iterative clustering algorithm, which obtains a direct effect on the formation of clusters. One of the problem in clustering is identifying the number of clusters for CCI. Such problems of the clustering technique have highlighted in many

recent research work. There are main approach such as average density-based initial cluster centers [Peng and Liu, 2018], initial centers based on dense and sparse neighbors [Huang et al., 2017], self-expressing the number of clusters and identifying initial centers from dense regions [Masud et al., 2018], locating initial cluster centers at the dense region of the dataset by representing the data points in KD-tree [Kumar and Reddy, 2017]. However, the computational complexity is the main drawback of KD-tree for the high dimensional dataset. The KDDcup99 (KDD'99) dataset is high volume, dynamic nature, and imbalance distributions of categories. Due to such nature, there is still no CCI method and no UFS method executed on this dataset.

Table 1: Comparison of related works

Work	Technique	KDD'99	Mode of training	UFS	CCI
Lin et al. [2015]	Clustering, k-NN, SVM	yes	Supervised, Unsupervised	no^*	no
Iglesias and Zseby [2015]	DTC, Bayes, kNN, ANN, SVM	yes	Supervised	no^*	n/a
Kang and Kim [2016]	Combinatorial optimization, MLP	yes	Supervised	no^*	n/a
Yin et al. [2017]	stream clustering	yes	Unsupervised	no	no
Bostani and Sheikhan [2017]	MI-BGSA, SVM	yes	Supervised	no^*	n/a
Al-Yaseen et al. [2017]	Clustering, ELM, SVM	yes	Supervised, Unsupervised	no	no
Roshan et al. [2018]	Adaptive clustering, ELM	yes	Supervised, Unsupervised	no	no
Al-Jarrah et al. [2018]	AdaBoost, Bagging, RF	yes	Semi-supervised	n/a	n/a
Choi et al. [2019]	Autoencoder, ANN	yes	Unsupervised	no	n/a
Meira et al. [2019]	Autoencoder, K-means, Isolation Forest, SVM, Nearest Neighbor, SCH	yes	Unsupervised	no	no

$no^* \leftarrow$ Supervised feature selection
$n/a \leftarrow$ not applicable

Kang and Kim [2016] proposed combinatorial optimization based optimal feature subset selection. The number of possible feature subsets from given n features is $(2^n - 1)$, their approach reached $(2^{41} - 1)$ complexity. They selected 20 features, which shown a better DR and poor FAR. Their method took more time to provide optimal subsets of features. Yin et al. [2017] proposed a stream clustering algorithm for anomaly detection. They completed execution in 10 rounds and reported system accuracy of each round. The best result has shown the DR (99%) and FAR (9.17%), while the average DR (91%) and average FAR (13.61%).

Al-Yaseen et al. [2017] preprocessed the training dataset into five sub-datasets: Normal-DS, DoS-DS, U2R-DS, Probe-DS, and R2L-DS using the supervised mode of training method, and reduced training set. The SVM

took more training time hence they approach to build small training dataset. Then, they apply modified K-means and basic K-means based on multilevel hybrid SVM and ELM with K=55. We have proposed UFS and CCI based clustering method, and executed on the unlabeled training set, which differs from the existing method. They reported that their method perform better for DR of normal and probe categories, while R2L (31.39%) and U2R (21.93%). Roshan et al. [2018] proposed an adaptive design of IDS based on ELM. A comparison of their method has obtained results in both modes of training, which shown in many cases, and the method has achieved better performance. They reported that their method achieved best DR (84%) and best FAR (3.02%).

Choi et al. [2019] developed a network IDS using an unsupervised learning method. Autoencoder model aims to reconstruct its input vectors as ANN-based learning. It learns to construct its input data consisting only normal data. They tested the performance on training data, which have normal (99%) and abnormal (1%) data, and reported the best result as accuracy (91.70%) and DR (84.68%). Meira et al. [2019] proposed anomaly detection using unsupervised learning techniques and explored six techniques as one-class classification; these techniques are K-means, Autoencoder, Isolation Forest, Nearest Neighbor, Support Vector Machine (SVM), Scaled Convex Hull (SCH). They discretized continuous features data using the Equal Frequency (EF) technique and normalized data using Zscore and MinMax technique. Finally, they reported limited overall performance. We have compared the proposed method to their methods and other unsupervised methods such as K-means (where K=5) and New K-means (where K=79 and merge them into 5), that achieves better performances.

Table 1 shows the comparison of IDS that performed on the KDD'99 dataset, and trained on both modes of training. There is still no unique method, that provides UFS and CCI based clustering method. The proposed work comprises the novel approach of UFS and CCI, which obtains the fast execution of clustering applications and provides better results. It indicates that the clustering method with initial centers is suitable for the detection system. Additionally, the feature selection method reduces the computational complexities of the system.

3. Mobile ad-hoc network security mechanism

The MANET is an infrastructure-less, temporary, self-organized, and dynamic network that transfers data packets to the destination node through multihop communication. It is more vulnerable due to wireless communication and dynamic nature [Bouhaddi et al., 2018]. This network mostly

applies for military services in the battlefield and search operations, where information security is the primary issue. An attacker can be internal or external that spread malicious information and harm the network resources. There are many types of attack methods in temporary networks, and each has a unique attacking nature.

A wormhole attack is a dominant network security threat that can not immune to cryptography and traditional security mechanisms. An attacker node attracts the request packets by advertising itself a part of the destination in minimum hop count (false information) and tunnel them to other ends [Tiruvakadu and Pallapa, 2018]. Malicious nodes are more active, where they strategically maintain locations and generate huge signals. Two malicious nodes create a wormhole tunnel when they are in the radio range and transmit packets without addressing themselves. When other malicious nodes are not in radio range, then it behaves as a blackhole attack. These malicious nodes also conceal their history and information. The proposed method confirms machine learning techniques effectively detect malicious data, which forwarded through the malicious node.

The wireless network is more versatile than the wired network. Sample labeling is one of the main stages of data preprocessing of supervised learning where each mistake can affect negatively on the dataset and overall performance of the predictive model. It becomes more difficult labeling the sample (or packet information) as normal or malicious in dynamic networks when attackers constantly changing their nature. Some attacking method changes its identity that also makes the labeling process difficult. Unsupervised mode of training model learns to unlabeled dataset [Carrasco and Sicilia, 2018], which is free from labeling cost and complexities. Therefore, unsupervised intrusion detection system is a suitable approach for MANETs.

4. IDS datasets

4.1. KDD

The KDD'99 dataset is most popular publicly available IDS dataset. It was generated by Defense Advanced Research Project Agency (DARPA) that contained normal and attack instances. MIT Lincoln Labs were prepared and managed Air force environment LANs with multiple attacks [Ring et al., 2019]. In 1999, it was merged with some new attacks and was named as KDDcup99 (KDD'99).

Table 2 presents the comparison of IDS datasets on different parameters [Sharafaldin et al., 2018]. The KDD'99 dataset satisfies maximum parameters and contains a maximum different type of attacks. It is an extremely

Table 2: Comparison of IDS datasets

Parameters	KDD'99	DEFCON	CAIDA	LBNL	Kyoto	ISCX2012	ADFA2013	CICIDS2017
Complete network	yes	no	yes	yes	yes	yes	yes	yes
Attack traffic	yes	yes	yes	yes	yes	yes	yes	yes
Normal traffic	yes	yes	yes	yes	yes	yes	yes	yes
Network interactions	yes	yes	no	no	yes	yes	yes	yes
Complete capture	yes	yes	no	no	yes	yes	yes	yes
Protocols	many	few	NS	few	many	many	many	many
Different attacks	39	NS	NS	NS	17	4	12	14
Meta data	yes	no	yes	no	yes	yes	yes	yes
Features set	yes	no	no	no	yes	no	no	yes

NS ← Not Specified

imbalanced dataset, and the test set contains 17 more attacks than a training set (more details in Table 3, 4, 5), which is more challenging to achieve better performance. This dataset contains three sub-datasets namely KDDcup dataset, KDD 10% dataset, and KDD corrected dataset. KDDcup is high volume dataset, that contains 4,898,431 (attacks 3,925,650 and normal 972,781) instances [Tavallaee et al., 2009]. The rest of sub-datasets details are in Table 3 as training and testing dataset.

Table 3: Categories and number of attacks of each category.

Category	Attack name	Training		Testing			
		KDD 10% dataset		KDD corrected dataset			
		Instances	Total	Instances	New Attack	Instances	Total
DoS	back	2203	391458	1098	apache2	794	229853
	land	21		9	mailbomb	5000	
	neptune	107201		58001	processtable	759	
	pod	264		87	udpstorm	2	
	smurf	280790		164091			
	teardrop	979		12			
Probe	ipsweep	1247	4107	306	mscan	1053	4166
	nmap	231		84	saint	736	
	portsweep	1040		354			
	satan	1589		1633			
R2L	ftp_write	8	1126	3	named	17	16347
	guess_passwd	53		4367	sendmail	17	
	imap	12		1	snmpgetattack	7741	
	multihop	7		18	snmpguess	2406	
	phf	4		2	worm	2	
	spy	2		-	xlock	9	
	warezclient	1020		-	xsnoop	4	
	warezmaster	20		1602	httptunnel	158	
U2R	buffer_overflow	30	52	22	ps	16	70
	loadmodule	9		2	sqlattack	2	
	perl	3		2	xterm	13	
	rootkit	10		13			

Table 3 provides detail of attacks and their categories. It shows four

categories of attacks namely DoS, R2L, U2R and Probe. The training dataset contains 22 attacks, while testing dataset contains 20 same attacks and 17 new attacks. Here, we have considered Httptunnel attack in R2L category as in paper [Aburomman and Reaz, 2016], whenever, some paper [Al-Yaseen et al., 2017] has included it in U2R category.

Table 4: New attack instance distributions of test set.

Category	New attack	Existing attack	Total
DoS	6555 (02.85%)	223298 (97.15%)	229853
Probe	1789 (42.94%)	2377 (57.06%)	4166
R2L	10354 (63.34%)	5993 (36.66%)	16347
U2R	31 (44.29%)	39 (55.71%)	70
Total	18729 (07.48%)	231707 (92.52%)	250436

Table 4 shows the instance distribution of new attack, which is not present in the training set. It also shows the existing attack instances those are present in the training set as well as the test set. Test dataset contains 7.48% new attack instances, which are maximum in R2L and minimum in DoS category. DoS attack instance distribution is much higher than other attacks, that reduces the overall new attack distribution. However, most of attacks contain approx 50 percent new attack instances except for DoS that resist achieving high DR mainly for unsupervised IDSs.

Table 5: Instance details of training and testing dataset

Category	Training KDD 10% dataset	Testing Corrected dataset
Normal	97278 (19.69%)	60591 (19.48%)
DoS	391458 (79.24%)	229853 (73.90%)
Probe	4107 (0.83%)	4166 (1.34%)
R2L	1126 (0.23%)	16347 (5.26%)
U2R	52 (0.01%)	70 (0.02%)
Total	494021	311027

Table 5 shows a detail distribution of instances of training and testing dataset, where available instances in each category are shown huge gaps among categories. In this table, it also shows the percentage of instance distribution, that a large difference among categories. The cluster size and shape depends on available similar behavior objects in the dataset. These datasets contain huge redundant instances, that are available mostly in attacks. The proposed method has executed on 494,021 instances for training

and 311,027 instances for testing. There are two invalid records found in the test set, those serial numbers are 136489 and 136497, that contains invalid value ICMP as their service value feature [Tavallaee et al., 2009].

This is publicly available and the most applicable IDS dataset. Some effective IDSs have executed on the subset of the dataset, that represents the complete instances. Recently, the modified K-means has used KDD'99 dataset, which five training datasets have constructed from available samples as one training set for each category [Al-Yaseen et al., 2017]. Unsupervised method [Roshan et al., 2018] has executed on the training sets of 12,000 and 20,000 samples, while they updated every iteration by 10, 20 and 30 percent of the size of training sets.

4.2. CICIDS2017

The Canadian Institute of Cybersecurity has generated the IDS dataset, namely CICIDS2017 (in July 2017) and CSE-CIC-IDS2018 (in Feb. 2018). Both datasets contain the same feature set [Prasad et al., 2020], attacks, and attacking nature. The main difference between them the CICIDS2017 is captured network traffic on relatively a small emulated network environment over five days [Sharafaldin et al., 2018, Ring et al., 2019]; while, the CSE-CIC-IDS2018 is captured network traffic and log files on large emulated network environment over 18 days.

Table 6: Data details and distribution

Sub-dataset	Class	Data samples		Total
		Training	Testing	
Tuesday	BENIGN	345635 (96.89%)	86439 (96.92%)	445909
	FTP Patator	6361 (1.78%)	1577 (1.77%)	
	SSH Patator	4731 (1.33%)	1166 (1.31%)	
Wednesday	BENIGN	351960 (63.51%)	88071 (63.57%)	692703
	Dos slowloris	4632 (0.84%)	1164 (0.84%)	
	Dos slowhttptest	4369 (0.79%)	1130 (0.82%)	
	Dos Hulk	184934 (33.37%)	46139 (33.3%)	
	Dos GoldenEye	8259 (1.49%)	2034 (1.47%)	
	Heartbleed	8 (0.001%)	3 (0.002%)	
Thursday Morning	BENIGN	134566 (98.73%)	33620 (98.67%)	170366
	Web Attack-Brute Force	1181 (0.87%)	326 (0.96%)	
	Web Attack-XSS	530 (0.39%)	122 (0.36%)	
	Web Attack-sql injection	16 (0.01%)	5 (0.01%)	
Thursday AfterNoon	BENIGN	230857 (99.99%)	57709 (99.98%)	288602
	Infiltration	25 (0.01%)	11 (0.02%)	
Friday Morning	BENIGN	151249 (98.97%)	37818 (98.99%)	191033
	Bot	1577 (1.03%)	389 (1.01%)	
Friday AfterNoon-DDoS	BENIGN	147312 (81.57%)	36598 (81.06%)	225745
	DDoS	33284 (18.43%)	8551 (18.94%)	
Friday AfterNoon-PortScan	BENIGN	101895 (44.46%)	25642 (44.76%)	286467
	PortScan	127279 (55.54%)	31651 (55.24%)	

Table 6 presents the details of the CICIDS2017 dataset. It has eight sub-datasets; where, Monday contains only BENIGN samples and rest details in the table. It shows that the dataset is highly imbalanced among their categories, which resist achieving satisfactory performances for unsupervised IDSs. In this execution, we have applied the holdout validation method and unlabeled training set to learn the system. This method randomly splits samples of sub-datasets into a training set (80%) and test set (20%). There is still no unsupervised intrusion detection and unsupervised feature selection performed on this dataset.

4.3. Wormhole dataset

Data collection from real-world-enterprise of MANETs for IDS is not easy work. Therefore, we simulated wormhole attack for the MANET environment using Network Simulator (NS-3), that is presented in [Prasad et al., 2019]. Simulation is performed on 20 normal and 5 malicious nodes, 250-meter radio range, random movement of nodes, 1000*1000 m^2 topology space, and 300 seconds simulation time, which has generated high volume dataset. Fig. 1 shows the steps of wormhole dataset generation process such as wormhole attack simulation, packet capture file, featuring data, and collection of data into the database.

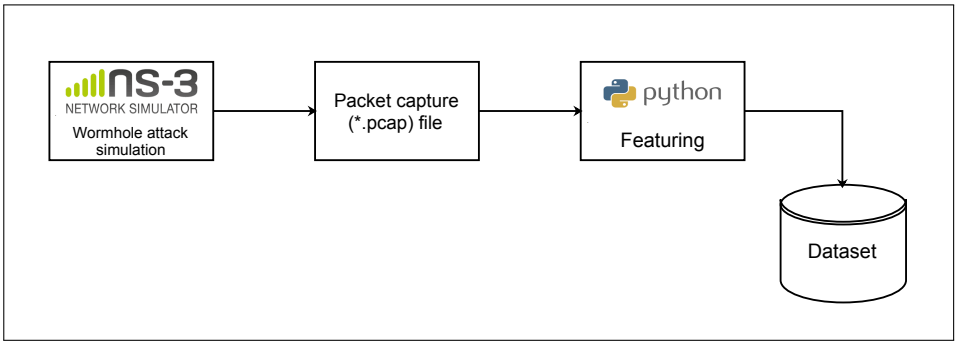


Figure 1: Wormhole dataset

This dataset contains 637,862 (152,144 normal and 485,718 malicious) samples, that are collected on 20 distinguish features of ad-hoc network. We have preprocessed the wormhole dataset by transferring symbolic value to numerical value using the label encoding method and normalizes numerical value into a well proportionate range. This preprocessed dataset contains many duplicate samples. We have executed the proposed method on only nonredundant (201,616) samples; whenever, detection methods in [Prasad et al., 2019] are executed on original (637,862 samples) dataset. Further, it has divided into 141,131 (70%) for training set and 60,485 (30%) for testing,

where test set contains 14,499 (24%) normal and 45,986 (76%) malicious information.

5. Proposed method

The following subsections elaborate proposed method. It starts with the selection of benchmark IDS dataset and data normalization method. The next step involves unsupervised feature selection that select best features. These features reduce dimensionality and redundant data. Afterward, the

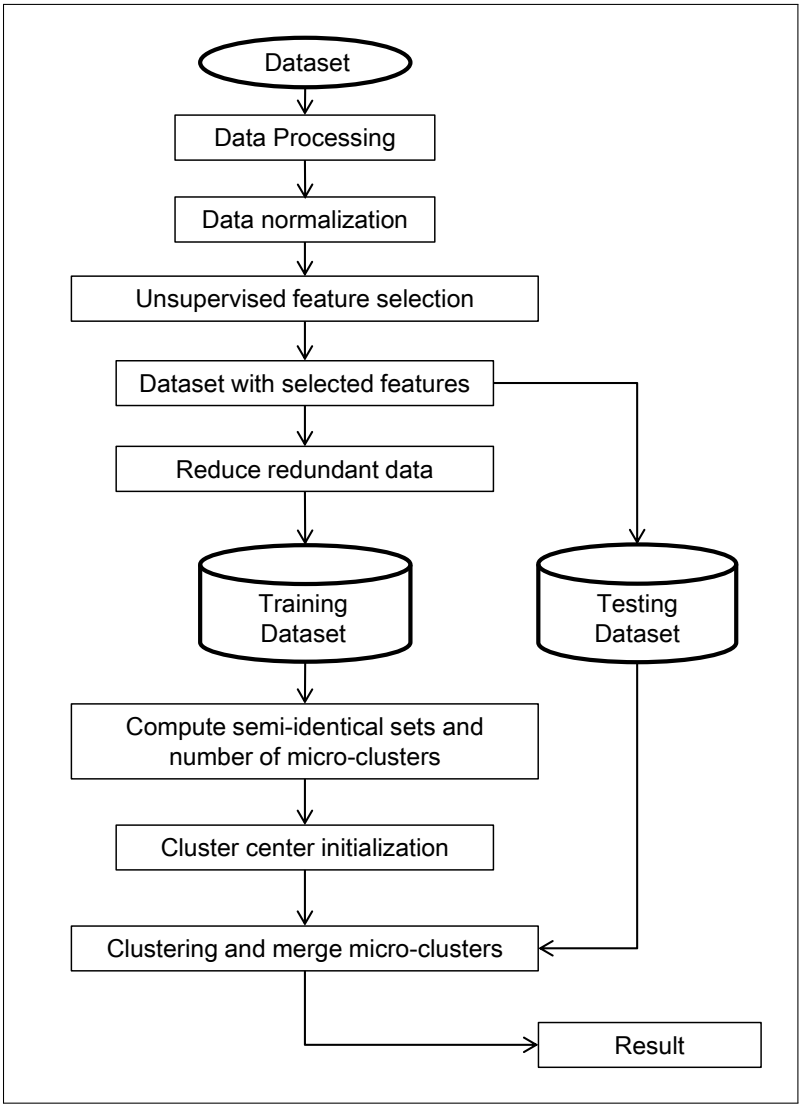


Figure 2: Diagrammatic representation of proposed method

next step creates sets of semi-identical instances. These sets are used to compute initial cluster centers. Subsequent steps describe the structure of sets and initialization of centers. The main step executes data clustering which carried out following changes : (1) computation of clusters range, (2) cluster center initialization, (3) number of clusters ($K \gg c$). Finally, it merges minimum distance micro-clusters and computes performance of the system. The unsupervised learning model is used on supervised data split into the train-test sets for validation and the performance metrics. The model has not undergone any prior training. It is presented with an unlabeled and uncategorized training dataset, and the learning procedure finds similarity between training samples and puts similar items into the same cluster. Fig. 2 shows diagrammatic representation of the proposed method.

5.1. Data normalization

Data normalization is one of the main steps which transferring symbolic value into a numerical value. Where each value of the feature requires to scale down into the well proportionate range [Ambusaidi et al., 2016]. We have applied the label encoding method, that provides numbers to distinct symbolic (string) objects in a sequence and normalizes them from 0 to 1 (using Eq. 1).

$$x_{ij} = \frac{\chi_{ij} - min(\chi_j)}{max(\chi_j) - min(\chi_j)}, \tag{1}$$

where, x_{ij} is normalized value of χ_{ij}. $Min(\chi_j)$ represents the minimum value and $max(\chi_j)$ is the maximum value of the j^{th} feature. This normalization eliminates greater deviation and also the bias of features. It applies on both training and test set with same min and same max value.

5.2. Unsupervised feature selection

The UFS reduces redundant and irrelevant features, and select non-redundant features of unlabeled dataset [Wu et al., 2016]. These redundant features can create problems such as increasing complexities, computations, deviate actual clusters, support wrong decisions, etc. It is more challenging task to select a subset of more informative features for unlabeled dataset, that can vanish essential information. Feature score can compute using entropy for unlabeled dataset by calculating histogram of the feature. However, the proposed method selects the only most frequent object to compute the feature score that differs from entropy. When the feature contains every entry the same or every entry different that has negligible contribution, the proposed method selects features that score within threshold range. This method reduces only negligible contribution features from any datasets.

Table 7: Features of KDD'99

Feature		Feature name	Type	eff	rel	status
Basic features	f1	duration	integer	small	negligible	yes
	f2	protocol_type	string	small	low	yes
	f3	service	string	small	strong	yes
	f4	flag	string	high	strong	yes
	f5	src_bytes	integer	small	low	no
	f6	dst_bytes	integer	small	strong	no
	f7	land	binary	medium	low	no
	f8	wrong_fragment	integer	high	strong	no
	f9	urgent	integer	medium	negligible	no
Content features	f10	hot	integer	high	low	yes
	f11	num_failed_logins	integer	high	negligible	no
	f12	logged_in	binary	high	low	yes
	f13	num_compromised	integer	high	low	no
	f14	root_shell	binary	high	low	no
	f15	su_attempted	binary	high	low	no
	f16	num_root	integer	high	negligible	no
	f17	num_file_creations	integer	high	low	no
	f18	num_shells	integer	high	low	no
	f19	num_access_files	integer	high	negligible	no
	f20	num_outbounds_cmds	integer	high	negligible	no
	f21	is_hot_login	binary	high	negligible	no
	f22	is_guest_login	binary	high	low	no
Traffic features	f23	count	integer	medium	strong	yes
	f24	srv_count	integer	medium	low	yes
	f25	serror_rate	real	high	strong	yes
	f26	srv_serror_rate	real	high	strong	yes
	f27	rerror_rate	real	high	low	yes
	f28	srv_rerror_rate	real	high	strong	yes
	f29	same_srv_rate	real	medium	strong	yes
	f30	diff_srv_rate	real	medium	negligible	yes
	f31	srv_diff_host_rate	real	medium	negligible	yes
	f32	dst_host_count	integer	medium	strong	yes
	f33	dst_host_srv_count	integer	medium	strong	yes
	f34	dst_host_same_srv_rate	real	medium	strong	yes
	f35	dst_host_diff_srv_rate	real	medium	strong	yes
	f36	dst_host_same_src_port_rate	real	medium	negligible	yes
	f37	dst_host_srv_diff_host_rate	real	medium	negligible	yes
	f38	dst_host_serror_rate	real	high	strong	yes
	f39	dst_host_srv_serror_rate	real	high	strong	yes
	f40	dst_host_rerror_rate	real	high	strong	yes
	f41	dst_host_srv_rerror_rate	real	high	low	yes

$$\psi_j = \frac{max\Big(\sum_{v=1}^{n_j}\sum_{i=1}^{N} 1|\gamma_{vj} = x_{ij}\Big)}{N}, \tag{2}$$

where, ψ_j is feature score, $\gamma_j = distinct(x_j)$, and $n_j = count(\gamma_j)$ of j^{th} feature. Eq. 2 computes the count of occurring each object and selects the max count of the feature. Then, the ratio of max count and total samples provide the feature score.

$$f_j = \begin{cases} 0, & if\ \psi_j \approx 1\ or\ \psi_j \approx 0 \\ 1, & otherwise \end{cases} \tag{3}$$

Eq. 3 indicates zero for not selected and one for selected features, when $\psi_j > 0.995$ or $\psi_j < 0.005$ follows its approximate value.

Table 7 contains the detail information of features of KDD'99 dataset. First four columns contain features type, features sequence, feature name and data type. Column 5 (eff) shows the effort to generate the feature, and Column 6 (rel) contains relevance of features [Iglesias and Zseby, 2015]. In terms of eff, features need a simple inspection of header fields as simple effort, need comparison as a medium effort, and transforming into another form as high effort. Iglesias et al. computed feature ranking and weighting using four different methods that show the contribution of features as negligible, low, and strong relevance [Iglesias and Zseby, 2015]. The final column contains the status of feature using the proposed UFS. In terms of relevance, selected features are 5 features (negligible contribution), 6 features (weak contribution), and 14 features (strong contribution). Similarly, in terms of effort to generate features that are 3 features (small), 11 features (medium), and 11 features (high). The comparative analysis of the proposed UFS to the existing method has shown a better subset of features, that contains strong relevance and medium effort.

Table 8: Selected features of KDD'99

No. of features	Features
25	f1, f2, f3, f4, f10, f12, f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f35 f36, f37, f38, f39, f40, f41

Table 8 contains selected features which is subset of 25 features. These features reduce dimensionality as well as redundant samples of the dataset. A subset of the dataset contains 81,798 non-redundant samples, which is

Table 9: An example information table for unsupervised feature selection and formation of semi-identical sets.

U	duration	protocol	packet_size	flag	header_length	hop_count
x_1	0.02	UDP	128	0	20	2
x_2	0.02	AODV	64	0	24	2
x_3	0.05	AODV	64	1	24	4
x_4	0.02	AODV	64	0	24	2
x_5	0.02	AODV	32	1	20	2
x_6	0.05	AODV	64	1	24	4
x_7	0.05	UDP	128	0	20	3
x_8	0.05	UDP	64	1	24	3
x_9	0.02	AODV	32	0	20	2
x_{10}	0.02	UDP	32	1	20	2
x_{11}	0.05	AODV	64	1	20	4
x_{12}	0.02	AODV	128	0	24	2

only 16% samples of the training (KDD 10%) dataset. This training subset increases the performance of the system and decreases complexities.

We illustrate UFS method on example Table 9 using Eq. 2 and 3, that contains 12 samples (N = 12) and six features.

(i) compute for duration feature, $\gamma_{duration} = \{0.02, 0.05\}$, $n_j = 2$, $\psi_{duration} = \frac{max(7,5)}{12}$, $\psi_{duration} = 0.583$

(ii) protocol feature, $\gamma_{protocol} = \{\text{UDP, AODV}\}$, $n_j = 2$, $\psi_{protocol} = \frac{max(4,8)}{12}$, $\psi_{protocol} = 0.667$

(iii) packet_size feature, $\gamma_{packet\_size} = \{128, 64, 32\}$, $n_j = 3$, $\psi_{packet\_size} = \frac{max(3,6,3)}{12}$, $\psi_{packet\_size} = 0.5$

(iv) flag feature, $\gamma_{flag} = \{0, 1\}$, $n_j = 2$, $\psi_{flag} = \frac{max(6,6)}{12}$, $\psi_{flag} = 0.5$

(v) header_length feature, $\gamma_{header\_length} = \{20, 24\}$, $n_j = 2$, $\psi_{header\_length} = \frac{max(6,6)}{12}$, $\psi_{header\_length} = 0.5$

(vi) hop_count feature, $\gamma_{hop\_count} = \{2, 4, 3\}$, $n_j = 3$, $\psi_{hop\_count} = \frac{max(7,3,2)}{12}$, $\psi_{hop\_count} = 0.583$

This computation provides feature score of features {duration, protocol, packet_size, flag, header_length, hop_count} as {0.583, 0.667, 0.5, 0.5, 0.5, 0.583} which are within the threshold ($\psi_j > 0.005$ and $\psi_j < 0.995$) range. Then, UFS method considers as non-redundant features.

5.3. Structure of set

Only dissimilar instances maintain a distance that is calculated using Euclidean distance. This section introduces a set of instances that approx-

imately maintain minimum distance. Here, we have randomly divided features into approximately two equal subsets. A subset of features represents new information system and computes semi-identical subsets of instances. It gives many sets as micro-clusters that further compute cluster center and cluster range for clustering. Let information system (IS) : T=(U,A) from dataset, then $B_1 \subset A$ and $B_2 \subset A$ are associated an equivalence relation.

$$IND(R) = \Big\{(x, y) \in U^2 | \forall_b \in R(B_1), b(x) = b(y)\Big\}. \tag{4}$$

Eq. 4 computes $K = U/IND(\{B_1\}) = \{K_1, K_2, ..., K_n\}$, where, U is a non-empty finite set of instances and A is selected attributes. Here, B_1 and B_2 are two subsets of selected attributes (i.e., $B_1 \cap B_2 = \phi$ and $B_1 \cup B_2 = A$). r is an index of a set, whose length is greater than ten. The value of r is $1 \leq r \leq K$ and K is number of sets (micro-clusters). Instances x and y are indiscernible from each other by attributes in B_1 subset [Ghosh et al., 2016]. For every $b \in R(B_1)$ and $b : U \rightarrow V_b$, where V_b is called value set of b. Eq. 4 group instances into set those are similar by features in B_1. An attribute subset B_1 contains 13 features and subset B_2 contains rest of features of KDD'99 dataset, whereas each subset contains 29 features of CICIDS2017 dataset and each subset contains 10 features of Wormhole dataset.

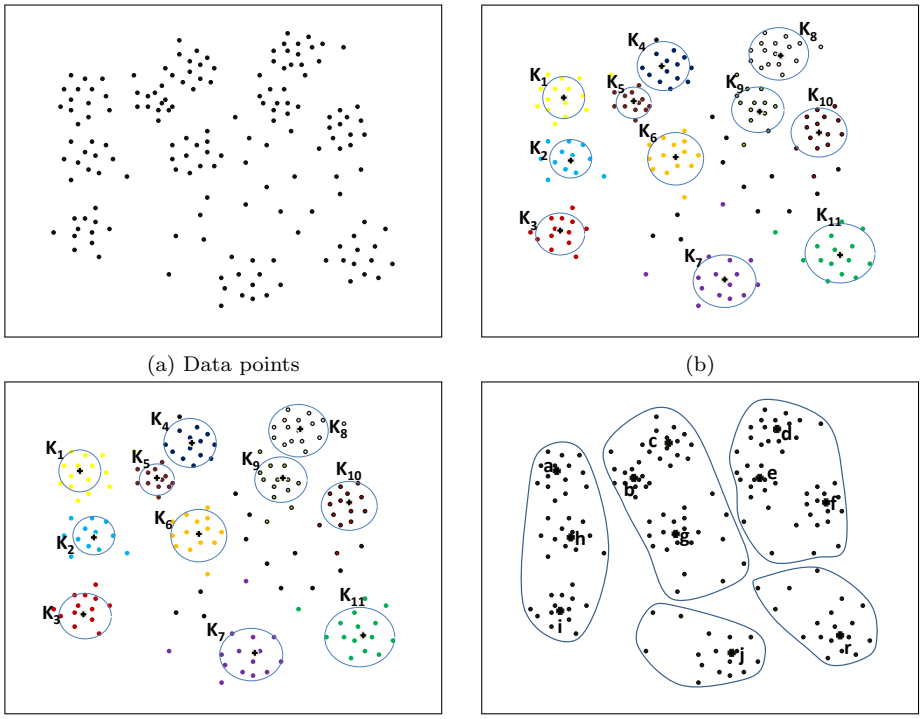
An example of semi-identical sets formation using the sample Table 9. A set of features is randomly divided into two subsets such as B_1 = {duration, protocol, hop_count} and B_2 = {packet_size, flag, header_length}. Eq. 4 computes U/IND($\{B_1\}$) = $\{\{x_1, x_{10}\}, \{x_2, x_4, x_5, x_9, x_{12}\}, \{x_3, x_6, x_{11}\}, \{x_7, x_8\}\}$, which indicates semi-identical sets as K = U/IND($\{B_1\}$) = $\{K_1, K_2, K_3, K_4\}$. From this example, $K_1 = \{x_1, x_{10}\}$, $K_2 = \{x_2, x_4, x_5, x_9, x_{12}\}$, $K_3 = \{x_3, x_6, x_{11}\}$, and $K_4 = \{x_7, x_8\}$. These sets are used to compute further steps using ($B_1 \cup B_2 = A$) features.

5.4. Cluster center initialization

Clustering is an important application of machine learning that includes unsupervised classification. An iterative clustering algorithm depends on initial cluster centers [Kumar and Reddy, 2017]. A selection of initial centers is extremely important that directly affect the formation of final clusters [Masud et al., 2018, Peng and Liu, 2018]. However, selection of outliers as initial cluster center can affect the shape and size of final clusters, and also increase iterations. The CCI based clustering resolves notable problems of basic clustering.

$$V_{r,j} = \frac{1}{|K_r|} \sum_{s=1} x_{s,j} | x_{s,j} \in K_r, \tag{5}$$

where, $V_{k,j}$ is initial cluster center and $|K_r|$ is the size (number of instances) of set K_r. Eq. 5 computes initial cluster center, which overcome all issues of the basic clustering algorithm. It reduces iterations and computational complexities.



(a) Data points (b)

(c) K-means gives final K-centers (d) Final c-clusters

Figure 3: An example of a sample dataset with five clusters and K sets. (b) Preliminary partitions it into sets K=$\{K_1, K_2, K_3, ..., K_{11}\}$ and center of each set marked by black solid plus sign. These sets of semi-identical instances measure initial centers, K-clusters and cluster range.

5.5. Clustering

The K-means clustering is one of the popular algorithm of machine learning, that ability to group similar data into clusters [Gong et al., 2018, Zhao et al., 2018]. However, it is susceptible to select starting point as cluster centers [Kumar and Reddy, 2017]. As the procedure of the algorithm, it randomly selects initial centers. Therefore, it does not guarantee to get unique clustering. The final cluster centroid may not be optimal that can converge into local optimal solutions. To solve this problem, we have proposed a

clustering method, which is based on an initial cluster center.

$$distance_{i,r} = \sqrt{\sum_{j=1}(x_{i,j} - V_{r,j})^2}, \tag{6}$$

$$avgDistance_r = \frac{1}{|K_r|} \sum_{s=1} distance_{s,k} | x_s \in K_r, \tag{7}$$

$$V_{r,j} = \frac{1}{|cluster_r|} \sum_{x_i \in cluster_r} x_{i,j}, \tag{8}$$

where, Eq. 6 computes Euclidean distance of cluster center V_k to instance x_i. Eq. 7 computes the average distance of instances present in same set, which is used for bound the range (i.e., avgDistance) of the cluster. Eq. 8 updates cluster centroids every iteration, here, range of cluster computes only once. A cluster obtains instances as data points only that are in the range of cluster. This algorithm has clustered data into K-clusters (i.e., K=79 for KDD, Table 15 contains K values of sub-datasets of CICIDS2017, and K=31 for Wormhole dataset), that avoids outliers as initial cluster centers and produce the optimal clusters.

Fig. 3 shows demonstration of clustering process on sample dataset, where Fig. 3a contains data points with five clusters. Fig. 3b presents K number of sets and initial centers from dense region. To maintain micro-clusters define range of micro-clusters by averaging data points of the semi-identical set. Fig. 3c shows final centers obtained using K-means clustering. Fig. 3d displays macro-clusters as arbitrary shape clusters.

6. Experiments

6.1. Performance measures

The prediction of test set is measured as confusion matrix. It each cell contributes to measure statistical parameters. Rows and columns of the confusion matrix carried out a quantitative representation of the information [Bostani and Sheikhan, 2017], where row indicates a prediction of classifier, whenever a column indicates the total instances of the class. The diagonal represents the correct prediction of the class.

$$DR = \frac{TP}{TP + FN}, \tag{9}$$

$$FAR = \frac{FP}{FP + TN}, \tag{10}$$

$$Precision = \frac{TP}{TP + FP}, \tag{11}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}, \tag{12}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{13}$$

$$W.Avg = \sum_{i=1}^{c} \frac{S_i}{S} * Performance_i. \tag{14}$$

TP Rate (TPR) or DR or Recall is a proportion of correct predicted positive cases and actual positive cases. FP Rate (FPR) or FAR is a proportion of false positive and actual negatives. Precision is a proportion of correctly predicted and number of predicted instances. F-measure is a harmonic mean of DR and precision. Accuracy is the proportion of correct prediction by total instances [Ambusaidi et al., 2016]. Weightage average defines proportional sum of performance, where $S = \{S_1 + S_2 + ... + S_c\}$ is total samples, c is number of class, and S_i is number of samples of i^{th} class. We have computed the overall performance of the system using Eq. 14 and related measures.

6.2. Performance analysis on KDD dataset

This section analyzes the performance of the proposed unsupervised intrusion detection system. We have executed the proposed method on different IDS datasets. The dimensionality reduction reduces the size of the dataset and selection of the initial cluster center reduces the iteration of the clustering algorithm. Moreover, the proposed method reduces the computational complexities and provides better accuracy.

Table 10: Confusion matrix of test dataset of KDD'99

Class	Normal	DoS	R2L	U2R	Probe
Normal	44157	6089	4204	16	203
DoS	847	207753	188	4	903
R2L	14683	15817	11596	20	3
U2R	285	113	338	30	1
Probe	619	81	21	0	3056

Table 10 contains outcome of the proposed method in the form of confusion matrix. It is used to compute different statistical parameters. An example for Normal category, that statistical parameters are the diagonal

of table as TP (i.e., 44157), addition of rows and columns except Normal category as TN is 239924 (i.e., 207753, 188, 4, 903; 15817, 11596, 20, 3; 113, 338, 30, 1; 81, 21, 0, 3056), addition of column values only Normal category except diagonal as FN is 16434 (i.e., 847, 14683, 285, 619), addition of row values only Normal category except diagonal as FP is 10512 (i.e., 6089, 4204, 16, 203), likewise compute for all categories. Respective equations compute DR, FAR, Precision and Accuracy are in Section 6.1 using TP, TN, FP, and FN. Table 11 shows the DR and Precision of DoS is high, and U2R is low. It shows accuracy 99% for U2R and Probe, whenever more than 92% of rest except R2L.

Table 11: Statistical parameters of test dataset of KDD'99

Parameters	Normal	DoS	R2L	U2R	Probe
TP	44157	207753	11596	30	3056
TN	239924	79232	264157	310220	306140
FP	10512	1942	30523	737	721
FN	16434	22100	4751	40	1110
DR	0.73	0.91	0.71	0.43	0.74
FAR	0.0419	0.0239	0.1035	0.0023	0.0023
Precision	0.81	0.99	0.28	0.04	0.81
Accuracy	0.92	0.93	0.89	0.99	0.99

Higher TP and TN indicate better system performance, while higher FP and FN degrade the system performance. In the context of Normal category, FP is the wrong prediction of the classifier, that allows attack to enter into the system. An opposite, FN is also the wrong prediction where Normal predicts as an attack, that increases the alert overhead. Minimum FAR shows better performance of the system like the proposed system, it gives low FAR of all categories except DoS. This system provides the weightage average (using Eq. 14) performance of respective statistical measures as overall performance.

Table 12: Detection rates (in %) of the proposed method and recent IDSs [Al-Yaseen et al., 2017], and the best detection rate of the different category is shown in boldface.

Category	Multi-level ELM	Multi-level SVM	Modified K-means	Proposed
Normal	96.64	97.83	**98.13**	72.87
DoS	96.83	**99.57**	99.54	90.39
R2L	10.84	31.60	31.39	**70.94**
U2R	23.68	16.23	21.93	**42.86**
Probe	84.93	80.94	**87.22**	73.36

Table 12 shows detection rates of recent algorithms [Al-Yaseen et al.,

2017]. The proposed method shows higher detection rate of U2R and R2L. While modified K-means performs better for normal and Probe categories, where it is based on multi-level hybrid ELM and SVM, whenever Multi-level SVM performs better for DoS category. Especially, the proposed method gives better detection rates of those categories that contain few samples.

Table 13: Performance comparisons (in %) of different methods

Technique	No. of features	Accuracy	DR	FAR
DTC [Iglesias and Zseby, 2015]	16	78.22	82.62	–
ANN [Iglesias and Zseby, 2015]	16	79.25	70.96	–
DTC [Iglesias and Zseby, 2015]	30	78.68	85.53	–
Bayes [Iglesias and Zseby, 2015]	41	57.01	99.16	–
ANN [Iglesias and Zseby, 2015]	41	77.74	92.79	–
MI-BGSA [Bostani and Sheikhan, 2017]	5	88.36	86.30	8.88
BGSA [Bostani and Sheikhan, 2017]	13	85.62	81.17	8.42
BPSO [Bostani and Sheikhan, 2017]	11	85.88	81.41	8.14
Multi-level ELM [Al-Yaseen et al., 2017]	41	93.83	95.02	3.36
Multi-level SVM [Al-Yaseen et al., 2017]	41	95.57	95.02	2.17
Basic K-means [Al-Yaseen et al., 2017]	41	91.88	92.13	9.16
Modified K-means [Al-Yaseen et al., 2017]	41	95.75	95.17	1.87
Autoencoder$^{\bullet}$ [Choi et al., 2019]	41	91.70	84.68	–
$K-means^{\bullet}$	25	51.44	26.15	4.37
New $K-means^{\bullet}$	25	88.13	80.16	5.30
Proposed$^{\bullet}$	25	93.07	84.70	2.73

$\bullet \leftarrow$ Unsupervised mode of training method.

Table 13 shows the overall performance of the system concerning Number of features, Accuracy, DR, and FAR. This clustering method is compared with DTC, Bayes, ANN, MI-BGSA, BGSA, BPSO, Multi-level ELM, Multi-level SVM, Basic K-means, and modified K-means clustering, while these have trained on the supervised mode of training. Iglesias and Zseby [2015] have trained and tested on three subsets (or sets) of features using five classifiers namely DTC, kNN, Bayes, ANN, and SVM. We have compared to at

Table 14: Performance comparison of unsupervised intrusion detection systems, and the best result is shown in boldface.

Technique	No. of features	DR	FAR
Initial model [Roshan et al., 2018]	41	0.74	0.0314
Standard ELM model [Roshan et al., 2018]	41	0.67	0.19
Updated model[Roshan et al., 2018]	41	0.84	0.0302
Proposed	25	**0.85**	**0.0273**

least one best result of each set. The proposed method is trained on the un-

supervised mode of training with selected 25 features, while, Autoencoder is trained on the unsupervised mode of training with 41 features, and K-means and New K-means are trained on unsupervised mode of training with 25 features. The comparative result shows that the proposed method performs better than the Autoencoder, K-means, New K-means methods.

Table 14 presents the performance comparison of unsupervised intrusion detection systems which have 41 features, while the proposed method has selected 25 features. The detection rate and system accuracy of the proposed method are better than unsupervised IDSs.

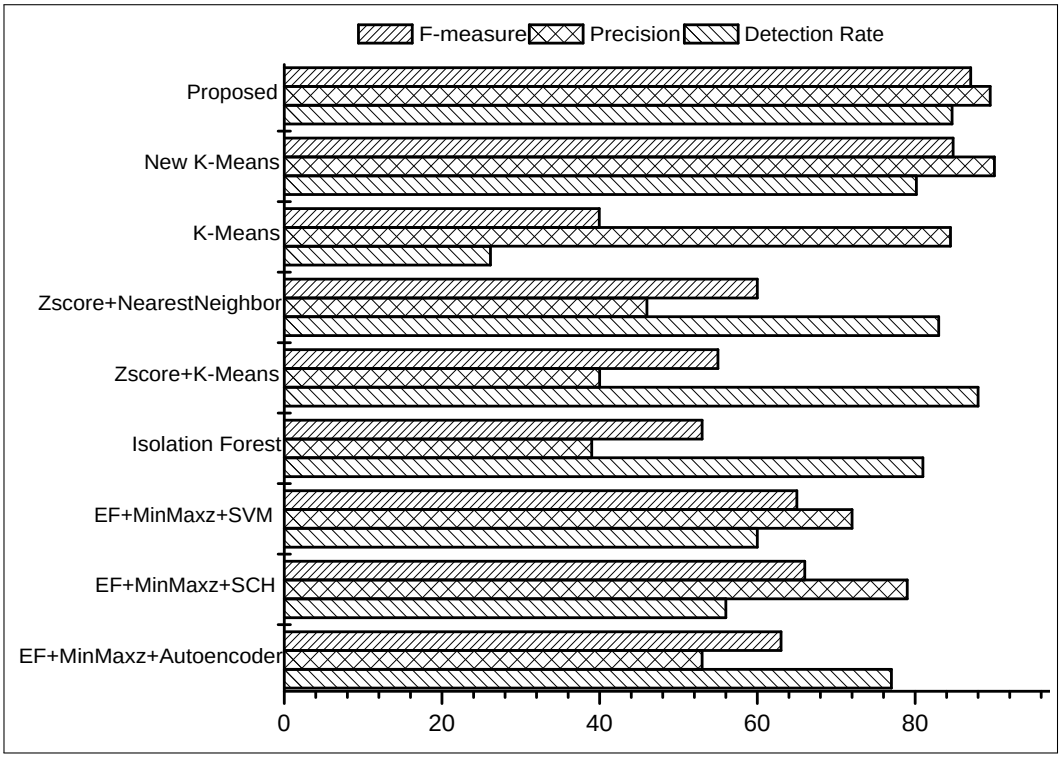
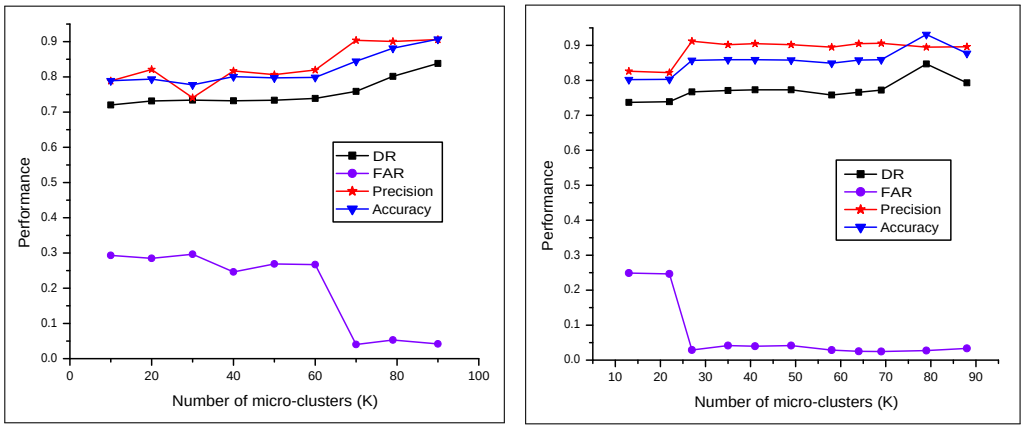


Figure 4: Performance (in %) of different unsupervised mechanisms on KDD dataset

Fig. 4 shows F-measure, DR, and precision of different unsupervised techniques, where F-measure is a harmonic mean of detection rate (DR) and precision. The K-means (with K=5), Isolation Forest, and Zscore+K-means show lower performance, whenever, Zscore + Nearest Neighbor, EF + MinMaxz + Autoencoder, EF + MinMaxz + SVM, EF + MinMaxz + SCH show average performance. New K-means and proposed method (for both K=79 and merge them into 5 clusters) achieve higher performance. Finally, UFS and CCI based micro-clustering mechanisms perform better than existing techniques.



(a) New K-means clustering (b) CCI based clustering

Figure 5: Performance of micro-clusters

Fig. 5 presents the performance comparison of micro-clustering methods on different statistical parameters as DR, FAR, Precision, and Accuracy. Fig. 5a shows the performance of New K-means clustering method. When the number of micro-clusters is more than 70, then, incline the performance of the system. The CCI based micro-clustering method has executed on different length of initial semi-identical sets as {600, 450, 350, 250, 200, 150, 110, 90, 70, 10, 5}, which provides number of micro-clusters as {13, 22, 27, 35, 41, 49, 58, 64, 69, 79, 88}. Fig. 5b shows the performance of the different number of micro-clusters. It achieves better performance on K=79. We have compared the proposed method to different unsupervised IDSs (Fig. 4), that obtained better performance.

6.3. Performance analysis on CICIDS2017 dataset

The CICIDS2017 and CSE-CIC-IDS2018 are benchmark datasets that overcome the shortcomings of outdated datasets. These are high volume and class-imbalanced datasets that are themself cause of challenges for unsupervised IDSs. The high volume dataset becomes a source of limitations as it consumes more overheads for processing. Especially, the clustering technique groups the data into an almost equal spatial size; whenever, the presence of data in benign class is much higher than attack such as Thursday AfterNoon dataset contains benign (99.99%) and attack (0.01%). These class-imbalanced data show higher FAR; whenever the system falsely predicts a small amount of samples. We have shown the ratio of falsely and correctly predicted (FP/TP) the clustering techniques. Similarly, accuracy represents the system performance as the ratio of correctly predicted and total samples. The proposed method is shown significant contributions for

Table 15: Details of preprocessed sub-datasets and identified sub-clusters

Sub-dataset	Training Samples	Nonredundant Samples	Test Samples	Weight of Test set	Number of Sub-clusters
Tuesday	356727	158480	89182	0.1938	24
Wednesday	554162	273014	138541	0.3010	28
Thursday Morning	136293	76991	34073	0.0741	13
Thursday AfterNoon	230882	111154	57720	0.1255	32
Friday Morning	152826	84110	38207	0.0830	19
Friday AfterNoon-DDoS	180596	83092	45149	0.0981	22
Friday AfterNoon-PortScan	229174	41362	57293	0.1245	21

unlabeled network traffic data and improved the system performances. Table 15 shows the training set and test set, while non-redundant samples are distinct samples of the training set. The weight of the test set computes the overall performances of the system (using Eq. 14). It is the proportional weight of test samples present in the set and total test samples. We have identified the number of sub-clusters using the proposed scheme and finally merge them into clusters.

Table 16: Category-wise statistical parameters of CCI based micro-clustering method

Sub-dataset	Class	TP	TN	FP	FN
Tuesday	BENIGN	86061	27	2716	378
	FTP Patator	27	87259	346	1550
	SSH Patator	0	87984	32	1166
Wednesday	BENIGN	86338	35230	15240	1733
	Dos slowloris	323	136796	581	841
	Dos slowhttptest	250	137146	265	880
	Dos Hulk	32162	90296	2106	13977
	Dos GoldenEye	663	135897	610	1371
	Heartbleed	0	138535	3	3
Thursday Morning	BENIGN	29465	337	116	4155
	Web Attack-Brute Force	259	30124	3623	67
	Web Attack-XSS	3	33346	605	119
	Web Attack-sql injection	0	34066	2	5
Thursday AfterNoon	BENIGN	57667	3	8	42
	Infiltration	3	57667	42	8
Friday Morning	BENIGN	37813	6	383	5
	Bot	6	37813	5	383
Friday AfterNoon-DDoS	BENIGN	34652	2361	6190	1946
	DDoS	2361	34652	1946	6190
Friday AfterNoon-PortScan	BENIGN	19490	17416	14235	6152
	PortScan	17416	19490	6152	14235

We have executed K-means, New K-means, and CCI based micro-clustering, which provide results in the form of confusion matrix. Table 16 presents the

Table 17: Category-wise performance of IDSs

Sub-dataset	Class	Performance	BRS	K-means	New K-means	CCI based
Tuesday	BENIGN	DR	0.9738	0.8970	0.9466	0.9956
		Accuracy	0.9746	0.8695	0.9330	0.9653
	FTP Patator	DR	1.0000	0.0000	0.5079	0.0171
		Accuracy	0.9977	0.9823	0.9392	0.9787
	SSH Patator	DR	0.4602	0.0017	0.0017	0.0000
		Accuracy	0.9930	0.8872	0.9807	0.9866
Wednesday	BENIGN	DR	0.9872	0.6199	0.8128	0.9803
		Accuracy	0.9874	0.6440	0.8281	0.8775
	Dos slowloris	DR	0.8656	0.3213	0.2895	0.2775
		Accuracy	0.9988	0.9341	0.9807	0.9897
	Dos slowhttptest	DR	0.9468	0.5354	0.5088	0.2212
		Accuracy	0.9994	0.9341	0.9779	0.9917
	Dos Hulk	DR	0.9892	0.6082	0.8545	0.6977
		Accuracy	0.9876	0.8547	0.8866	0.8839
	Dos GoldenEye	DR	0.9807	0.5088	0.3904	0.3260
		Accuracy	0.9996	0.9667	0.9839	0.9857
	Heartbleed	DR	1.0000	0.0000	1.0000	0.0000
		Accuracy	1.0000	0.8887	0.9701	0.9999
Thursday Morning	BENIGN	DR	0.9976	0.6370	0.8263	0.8764
		Accuracy	0.9960	0.6286	0.8276	0.8747
	Web Attack-Brute Force	DR	0.5634	0.0000	0.8589	0.7945
		Accuracy	0.9926	0.7640	0.9438	0.8917
	Web Attack-XSS	DR	0.4607	0.0164	0.0328	0.0246
		Accuracy	0.9933	0.8823	0.9314	0.9788
	Web Attack-sql injection	DR	0.5294	0.0000	0.6000	0.0000
		Accuracy	0.9995	0.9823	0.9446	0.9998
Thursday AfterNoon	BENIGN	DR	1.0000	0.9176	0.9748	0.9993
		Accuracy	0.9999	0.9176	0.9746	0.9991
	Infiltration	DR	0.9444	0.8182	0.2727	0.2727
		Accuracy	0.9999	0.9176	0.9746	0.9991
Friday Morning	BENIGN	DR	0.8705	0.8911	0.9267	0.9999
		Accuracy	0.8719	0.8820	0.9898	0.9898
	Bot	DR	1.0000	0.0000	0.3805	0.0154
		Accuracy	0.8719	0.8820	0.9898	0.9898
Friday AfterNoon-DDoS	BENIGN	DR	0.8486	0.8881	0.8188	0.9468
		Accuracy	0.8766	0.7722	0.7502	0.8198
	DDoS	DR	0.9990	0.2761	0.4568	0.2761
		Accuracy	0.8766	0.7722	0.7502	0.8198
Friday AfterNoon-PortScan	BENIGN	DR	0.9566	0.1099	0.8358	0.7601
		Accuracy	0.9808	0.6010	0.6790	0.6442
	PortScan	DR	1.0000	0.9987	0.5520	0.5503
		Accuracy	0.9808	0.6010	0.6790	0.6442

statistical parameters as TP, TN, FP, FN of CCI based micro-clustering method. A set of these parameters compute the performance of the system using the related equations. Similarly, we have computed the performance

of the rest of the clustering techniques; then, presented their results.

Table 17 shows the category-wise performance of the BRS and clustering techniques. These have been executed with the same training and testing sets that have only difference among them of their specific steps. This table shows the performances as detection rates and detection accuracy. The proposed CCI based method shows a better result than other clustering techniques. BRS in [Prasad et al., 2020] is a supervised method of intrusion detection that has worked on the same dataset; while, the proposed method is unsupervised training method. BRS has explicitly shown feature sets, attacks, data generation environment, and evaluated the qualitative realism of datasets. The comparative result has shown that the CICIDS2017 maintains high quality.

Table 18: Sub-dataset (or data-generation-day) wise weighted average performance of clustering techniques

Sub-dataset	Clustering Technique	DR	Precision	FP/TP	FAR	Accuracy
Tuesday	K-means	0.8695	0.9361	0.1501	0.9698	0.8717
	New K-means	0.9264	0.9560	0.0794	0.4805	0.9337
	Proposed	0.9653	0.9480	0.0360	0.9597	0.9658
Wednesday	K-means	0.6111	0.8072	0.6363	0.2083	0.7237
	New K-means	0.8136	0.8571	0.2291	0.1251	0.8524
	Proposed	0.8643	0.8675	0.1571	0.1996	0.8831
Thursday Morning	K-means	0.6286	0.9664	0.5908	0.9850	0.6309
	New K-means	0.8237	0.9867	0.2140	0.0770	0.8291
	Proposed	0.8725	0.9835	0.1462	0.2537	0.8752
Thursday AfterNoon	K-means	0.9176	0.9998	0.0898	0.1818	0.9176
	New K-means	0.9746	0.9997	0.0260	0.7271	0.9746
	Proposed	0.9991	0.9997	0.0009	0.7271	0.9991
Friday Morning	K-means	0.8821	0.9786	0.1338	0.9910	0.8820
	New K-means	0.9212	0.9837	0.0856	0.6140	0.9211
	Proposed	0.9899	0.9855	0.0103	0.9746	0.9898
Friday AfterNoon-DDoS	K-means	0.7722	0.7502	0.2950	0.6079	0.7722
	New K-means	0.7502	0.7720	0.3329	0.4746	0.7502
	Proposed	0.8198	0.7916	0.2199	0.5968	0.8198
Friday AfterNoon-PortScan	K-means	0.6009	0.7621	0.6641	0.4922	0.6009
	New K-means	0.6790	0.7145	0.4727	0.2912	0.6790
	Proposed	0.6441	0.6669	0.5524	0.3338	0.6442

Table 18 presents the performance of each subset of CICIDS2017 on different statistical parameters, such as DR, Precision, FP/TP, FAR, and Accuracy. The proposed method performs better for almost all subsets than clustering technique K-means and micro-clustering method New K-means. A micro-clustering technique is performed with the number of sub-clusters of respective subsets as the proposed method; finally, merge them into macro-clusters. Moreover, Table 19 shows the overall (or weighted average of test

Table 19: Overall performance of unsupervised IDSs on CICIDS2017

Clustering Technique	DR	Precision	F-measure	FP/TP	FAR	Accuracy
K-means	0.7380	0.8712	0.7991	0.3984	0.5496	0.7725
New K-means	0.8424	0.8882	0.8645	0.2021	0.3615	0.8559
Proposed	0.8800	0.8857	0.8828	0.1564	0.5371	0.8860

subsets) performance of the unsupervised IDSs, which is computed using the performances of sub-datasets or data generation days (in Table 18) and their weights (in Table 15). This computation carried out an additional statistical parameter F-measure as harmonic mean of detection rate and precision. Accuracy shows correctly predicted performance, and FP/TP falsely predicted the performance of the system. These statistical measures confirm that the proposed method performs better than unsupervised IDSs.

Table 20: Performance comparison of IDSs on CICIDS2017

Technique	No. of Features	DR	Precision	F-measure
Adaboost [Sharafaldin et al., 2018]	80	0.84	0.77	0.80
MLP [Sharafaldin et al., 2018]	80	0.83	0.77	0.79
QDA [Sharafaldin et al., 2018]	80	0.88	0.97	0.92
KNN [Sharafaldin et al., 2018]	80	0.96	0.96	0.96
RF [Sharafaldin et al., 2018]	80	0.97	0.98	0.97
ID3 [Sharafaldin et al., 2018]	80	0.98	0.98	0.98
BRS [Prasad et al., 2020]	40	0.96	0.96	0.96
$K-means^{\bullet}$	58	0.74	0.87	0.80
New $K-means^{\bullet}$	58	0.84	0.89	0.86
$Proposed^{\bullet}$	58	0.88	0.89	0.88

$\bullet \leftarrow$ Unsupervised mode of training method.

Table 20 summarises the comparative results of the proposed unsupervised IDS. It is compared to both modes of training methods, such as supervised and unsupervised IDSs. This table indicated the unsupervised techniques that are trained on unlabeled training sets. Whenever, supervised methods are trained on labeled training sets. The proposed method evaluates unsupervised feature selection, and it selects 58 significant features. The supervised methods are executed on extracted 80 features without assessing the significance of features except BRS [Prasad et al., 2020]. The proposed method is compared with both modes of the training method. It performs better than some supervised mode of training methods and all unsupervised techniques. Moreover, the proposed method confirms that it is suitable for classifying the behavior of unlabeled network traffic in benign and attacks

categories.

6.4. Performance analysis on wormhole dataset

New K-means clustering is similar to the basic clustering, that randomly selects initial cluster centers. The proposed method is CCI based clustering. Moreover, both clustering methods create micro-clusters and merge them into clusters. Table 21 shows the statistical parameters of clustering methods. Here, the number of micro-clusters is 31, and number of iterations of New K-means is 18, while the proposed method took average 10 iterations to form final micro-clusters.

Table 21: Statistical parameters of test dataset of Wormhole dataset

Parameters	New K-means			Proposed		
	Normal	Attack	W.Avg	Normal	Attack	W.Avg
TP	12894	43843	–	13304	43588	–
TN	43843	12894	–	43588	13304	–
FP	2143	1605	–	2398	1195	–
FN	1605	2143	–	1195	2398	–
DR	0.8893	0.9534	0.9380	0.9176	0.9479	0.9406
FAR	0.0466	0.1106	0.0953	0.0521	0.0824	0.0752
Precision	0.8574	0.9647	0.9390	0.8473	0.9733	0.9431
F-measure	0.8731	0.9590	0.9384	0.8810	0.9604	0.9413
Accuracy	0.9380	0.9380	0.9380	0.9406	0.9406	0.9406

Table 22: Performance comparison (in %) of detection methods on wormhole dataset, and the best result is shown in boldface.

Technique	DR	FAR	Precision	F-measure	Accuracy
Naive Bayes [Prasad et al., 2019]	93.12	**5.3**	94.0	93.4	93.06
SGD [Prasad et al., 2019]	93.12	**5.3**	94.0	93.4	93.08
New K-means	93.80	9.53	93.90	93.84	93.80
Proposed	**94.06**	7.52	**94.31**	**94.13**	**94.06**

Table 22 shows the performance of detection methods where the proposed method outperformed to the state-of-the-art method. The experimental results confirm that the CCI based clustering method performs better than basic clustering. Traditional approaches are also shown maximum 90% detection rates of wormhole attack in MANETs [Prasad et al., 2019]. This work confirms that machine learning algorithms perform better than traditional algorithms, and the proposed method performs better than existing detection methods.

7. Conclusion

This research work proposed a new clustering method for unsupervised intrusion detection. It is based on novel approach of unsupervised feature selection and initialization of the cluster center. The proposed method selects essential features, and the process of computation of semi-identical instances gives a number of micro-clusters, initial cluster centers, and a range of clusters. It avoids outliers as initial cluster center and more than one initial center from same cluster space. The quality of results for this proposal has been tested and compared to the state-of-the-art method. Mainly, the proposed unsupervised feature selection and cluster center initialization approach boost clustering method. This is trained and tested on KDD'99, CICIDS2017, and Wormhole dataset. Experimental results are shown better detection rate and system accuracy for attacks and non-attack classes.

The system is tested against existing methods and results are found to be encouraging. The implication of the proposed method is a demonstration of the fact that unsupervised feature selection and cluster center initialization method reduces training complexity. This method reduced the features that have negligible contributions. Experimental results confirm the performance of the proposed method is better than without feature reduction systems, and CCI based clustering is better than basic clustering method. The proposed method of unsupervised intrusion detection system can be used to provide security in MANETs, organizational and social areas where intruders are more active. The proposed work is convincing although it has some limitations like manually preprocessing, space and time complexities for MANETs. This system achieves promising performance that allows us to extend it and resolves these limitations. The extension may analyze complexities and accommodate the probabilistic approach for features score.

A. Features of Wormhole dataset

Feature	Feature name	Type
f1	duration	real
f2	protocol	string
f3	packet size	integer
f4	flag	integer
f5	header length	integer
f6	hop count	integer
f7	life time	integer
f8	message type	string
f9	destination sequence number	integer

Feature	Feature name	Type
f10	message sequence number	integer
f11	stream index	integer
f12	land	integer
f13	message transfer mode	binary
f14	number of neighbors	integer
f15	highest flow	integer
f16	average flow	real
f17	lowest flow	integer
f18	average hop count	integer
f19	number of failed connections	integer
f20	failed connection rate	real
f21	label	string

References

A. A. Aburomman and M. B. I. Reaz. A novel svm-knn-pso ensemble method for intrusion detection system. *Applied Soft Computing*, 38:360–372, 2016.

O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, and M. Al-Qutayri. Semi-supervised multi-layered clustering model for intrusion detection. *Digital Communications and Networks*, 4(4):277–286, 2018.

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*, 67:296–303, 2017.

M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan. Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE transactions on computers*, 65(10):2986–2998, 2016.

H. Bostani and M. Sheikhan. Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. *Soft computing*, 21(9):2307–2324, 2017.

M. Bouhaddi, M. S. Radjef, and K. Adi. An efficient intrusion detection in resource-constrained mobile ad-hoc networks. *Computers & Security*, 76: 156–177, 2018.

R. S. M. Carrasco and M.-A. Sicilia. Unsupervised intrusion detection through skip-gram models of network behavior. *Computers & Security*, 78:187–197, 2018.

H. Choi, M. Kim, G. Lee, and W. Kim. Unsupervised learning approach for network intrusion detection system using autoencoders. *The Journal of Supercomputing*, pages 1–25, 2019.

S. Ghosh, P. S. Prasad, and C. R. Rao. An efficient gaussian kernel based fuzzy-rough set approach for feature selection. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 38–49. Springer, 2016.

W. Gong, R. Zhao, and S. Grünewald. Structured sparse k-means clustering via laplacian smoothing. *Pattern Recognition Letters*, 112:63–69, 2018.

J. Huang, Q. Zhu, L. Yang, D. Cheng, and Q. Wu. Qcc: a novel clustering algorithm based on quasi-cluster centers. *Machine Learning*, 106(3):337–357, 2017.

R. Hyde, P. Angelov, and A. R. MacKenzie. Fully online clustering of evolving data streams into arbitrarily shaped clusters. *Information Sciences*, 382: 96–114, 2017.

F. Iglesias and T. Zseby. Analysis of network traffic features for anomaly detection. *Machine Learning*, 101(1-3):59–84, 2015.

S.-H. Kang and K. J. Kim. A feature selection approach to find optimal feature subsets for the network intrusion detection system. *Cluster Computing*, 19(1):325–333, 2016.

K. M. Kumar and A. R. M. Reddy. An efficient k-means clustering filtering algorithm using density based initial cluster centers. *Information Sciences*, 418:286–301, 2017.

W.-C. Lin, S.-W. Ke, and C.-F. Tsai. Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, 78:13–21, 2015.

G. Ma, Z. Xu, W. Zhang, and S. Li. An enriched k-means clustering method for grouping fractures with meliorated initial centers. *Arabian Journal of Geosciences*, 8(4):1881–1893, 2015.

M. A. Masud, J. Z. Huang, C. Wei, J. Wang, I. Khan, and M. Zhong. I-nice: A new approach for identifying the number of clusters and initial cluster centres. *Information Sciences*, 466:129–151, 2018.

J. Meira, R. Andrade, I. Praça, J. Carneiro, V. Bolón-Canedo, A. Alonso-Betanzos, and G. Marreiros. Performance evaluation of unsupervised techniques in cyber-attack anomaly detection. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 2019.

P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys & Tutorials*, 21(1):686–728, 2018.

L. Peng and Y. Liu. Attribute weights-based clustering centres algorithm for initialising k-modes clustering. *Cluster Computing*, pages 1–9, 2018.

M. Prasad, S. Tripathi, and K. Dahal. Wormhole attack detection in ad hoc network using machine learning technique. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2019.

M. Prasad, S. Tripathi, and K. Dahal. An efficient feature selection based bayesian and rough set approach for intrusion detection. *Applied Soft Computing*, 87:105980, 2020.

M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho. A survey of network-based intrusion detection data sets. *Computers & Security*, 86: 147–167, 2019.

S. Roshan, Y. Miche, A. Akusok, and A. Lendasse. Adaptive and online network intrusion detection system using clustering and extreme learning machines. *Journal of the Franklin Institute*, 355(4):1752–1779, 2018.

I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.

M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pages 1–6. IEEE, 2009.

D. S. K. Tiruvakadu and V. Pallapa. Confirmation of wormhole attack in manets using honeypot. *Computers & Security*, 76:32–49, 2018.

H. Wang, Y. Zhang, J. Zhang, T. Li, and L. Peng. A factor graph model for unsupervised feature selection. *Information Sciences*, 480:144–159, 2019.

Y. Wu, C. Wang, J. Bu, and C. Chen. Group sparse feature selection on local learning based clustering. *Neurocomputing*, 171:1118–1130, 2016.

T. Xie, P. Ren, T. Zhang, and Y. Y. Tang. Distribution preserving learning for unsupervised feature selection. *Neurocomputing*, 289:231–240, 2018.

C. Yin, S. Zhang, Z. Yin, and J. Wang. Anomaly detection model based on data stream clustering. *Cluster Computing*, pages 1–10, 2017.

Y. Zhao, Y. Ming, X. Liu, E. Zhu, K. Zhao, and J. Yin. Large-scale k-means clustering via variance reduction. *Neurocomputing*, 307:184–194, 2018.

P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo. Subspace clustering guided unsupervised feature selection. *Pattern Recognition*, 66:364–374, 2017.