Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025262
Article

Tech Science Press

An Efficient Intrusion Detection Framework in Software-Defined Networking for Cybersecurity Applications

Ghalib H. Alshammri[1,2], Amani K. Samha[3], Ezz El-Din Hemdan[4], Mohammed Amoon[1,4] and Walid El-Shafai[5,6,*]

[1]Department of Computer Science, Community College, King Saud University, Riyadh, 28095, Saudi Arabia
[2]Deanship of Scientific Research, Saudi Electronic University, Riyadh, Saudi Arabia
[3]Management Information System Department, College of Business Administration, King Saud University, Riyadh, 28095, Saudi Arabia
[4]Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf, 32952, Egypt
[5]Security Engineering Lab, Computer Science Department, Prince Sultan University, Riyadh, 11586, Saudi Arabia
[6]Electronics and Electrical Communications Eng. Depart., Faculty of Electronic Engineering, Menoufia University, Menouf, 32952, Egypt
*Corresponding Author: Walid El-Shafai. Email: walid.elshafai@el-eng.menofia.edu.eg
Received: 18 November 2021; Accepted: 24 January 2022

Abstract: Network management and multimedia data mining techniques have a great interest in analyzing and improving the network traffic process. In recent times, the most complex task in Software Defined Network (SDN) is security, which is based on a centralized, programmable controller. Therefore, monitoring network traffic is significant for identifying and revealing intrusion abnormalities in the SDN environment. Consequently, this paper provides an extensive analysis and investigation of the NSL-KDD dataset using five different clustering algorithms: K-means, Farthest First, Canopy, Density-based algorithm, and Exception-maximization (EM), using the Waikato Environment for Knowledge Analysis (WEKA) software to compare extensively between these five algorithms. Furthermore, this paper presents an SDN-based intrusion detection system using a deep learning (DL) model with the KDD (Knowledge Discovery in Databases) dataset. First, the utilized dataset is clustered into normal and four major attack categories via the clustering process. Then, a deep learning method is projected for building an efficient SDN-based intrusion detection system. The results provide a comprehensive analysis and a flawless reasonable study of different kinds of attacks incorporated in the KDD dataset. Similarly, the outcomes reveal that the proposed deep learning method provides efficient intrusion detection performance compared to existing techniques. For example, the proposed method achieves a detection accuracy of 94.21% for the examined dataset.

Keywords: Deep neural network; DL; WEKA; network traffic; intrusion and anomaly detection; SDN; clustering and classification; KDD dataset

1 Introduction

Recently, the SDN has been developed to be one of the talented answers for changing the future of global networks and the Internet [1–3]. The new emerging technology SDN detaches the data and control planes separately. The control plane can manage the complete network security concerns [4–7].

Deep learning (DL) is based on the extraction process of big data with complicated structures, so it is considered a type of machine learning model [8,9]. It is valuable in the case of learning from the huge size of unsupervised data [10,11]. The employment of deep learning achieves several advantages like higher classification performance and enhanced quality of produced samples. These benefits can realize superior machine learning results in applications like automatic natural language processing, speech recognition, computer vision, image recognition, and bioinformatics [12,13]. Furthermore, there are numerous valuable attributes of the learned abstract descriptions using deep learning as pretty straightforward linear models can perform efficient performance [14].

The network traffic, suspicious activities, and network administration can be controlled through the utilization of the IDS (Intrusion Detection System) [15,16]. The IDS framework involves an analysis engine, sensors, and a reporting system. SDN-based security is an important model to control the malicious flows in SDN switches [17,18]. Therefore, with the increasing amount of severe attacks and threats against a variety of computing systems such as a computer, network, cloud, and internet of things, the study of IDS has received a lot of attention from several researchers in the last decades in the security area [19]. The spiteful attacks such as theft of information can be detected using the IDS model [20,21]. The NIST (National Institute of Standards and Technology) [22] describes the incursion or intrusion as "an effort to negotiate the CIA (Availability, Integrity, and Confidentiality), or to evade the protection procedures of a network or computer".

The clustering-based IDS schemes are efficient compared to the traditional IDS schemes for detecting unknown attacks [23]. Furthermore, the KDD (Knowledge Discovery in Databases) is employed to find out and extract valuable information from huge relational databases. Therefore, data mining is important for determining significant non-intuitive patterns and correlations for obtaining superior knowledge from small data [24,25].

Intrusion detection using the unlabeled data technique, which is known as clustering, can group related records in the same clusters. Then, the common distance metrics on these clusters can be used to determine the anomalies [26]. The clustering process is an unsupervised learning process that deals with unlabeled data [27]. The machine learning (ML)-based intrusion detection analysis can help in detecting abnormal behavior in a network [28]. Nevertheless, several algorithms are available for intrusion detection, but their performance is needed to improve. One subset of ML methods is DL. Deep learning gained a reputation due to its perspective for machine learning.

For this reason, DL methods have been employed to identify some pattern types or cataloging. Furthermore, the application of deep learning can help to improve the accuracy and performance of IDS frameworks [29]. To perform and understand data analysis for this work, there are several data mining process models as the following [30]:

- **KDD (Knowledge Discovery in Databases):** The KDD is a process of how specialists can extract patterns and insights from data. It consists of five stages: Interpretation/Evaluation, Data Mining, Transformation selection, and Preprocessing. This model is followed in this research to achieve the paper's task of constructing an intrusion detection system.

- **SEMMA (Sample, Explore, Modify, Model, and Access):** The SEMMA model has a similar structure to KDD, nevertheless as it does not focus as deeply on data-specific phases, it is easier to apply to general data analysis tasks.
- **CRISP-DM (Cross-Industry Standard Process for Data Mining):** This model was formerly established in IBM for data mining tasks, finds it useful for almost all several data analytics projects.

This research paper investigates, discusses, and analyses five clustering algorithms: K-means, Farthest First, Canopy, Density-based algorithm, and Exception-maximization (EM), using the WEKA software to compare extensively between these five algorithms. The primary objective of this work is to analyze the NSL-KDD dataset. Furthermore, this paper presents an SDN-based intrusion detection system using a DL model with the NSL-KDD dataset.

In conclusion, the contribution of this paper is as follows:

- Provide an extensive comprehensive analysis of intrusion network traffic data via the NSL-KDD dataset. The utilized dataset is clustered into normal and four major attack categories.
- Develop a deep learning model for building a proficient SDN-based IDS.
- Conducting a systematic evaluation of the intrusion detection based on ML performances over the NSL-KDD dataset with the proposed model.
- The results provide a comprehensive analysis and a clear, reasonable study of various kinds of attacks incorporated in the NSL-KDD dataset. Similarly, the findings reveal that the suggested deep learning technique provides efficient IDS performance compared to existing procedures.

The remainder of this paper is constructed as follows. Section 2 presents a brief discussion of IDS, intrusion detection methodologies, deep learning, and SDN-based IDS, while the recent related works are investigated and provided in Section 3. Section 4 illustrates the suggested intrusion data-based clustering and detection scenarios, while the simulation outcomes and comparative analysis are presented in Section 5. Ultimately, the conclusion is stated in Section 6.

2 Preliminary Knowledge

This section presents a brief discussion of intrusion detection systems, intrusion detection methodologies, deep learning, and SDN-based IDS.

2.1 Intrusion Detection Systems

An IDS is considered a software or hardware system that checks a network's activities or system for strategy malicious actions and creates reports to the management and administrative system. The key task of intrusion prevention and detection systems (IPS/IDS) is to recognize the attacks and intrusions. Additionally, they are used to prevent persons and exhibit document warnings from violating security procedures and recognize problems with security guidelines.

The IDS/IPS has become a crucial matter for security purposes. Numerous techniques can be employed for the intrusion detection process [31]. They are used to protect computer networks in the presence of several attacks. On the other hand, it may also discard the packets or dismiss the connection. The intrusion systems can be classified as follows [32,33].

- **Host-based IDS (HIDS):** The HIDS is exploited to protect a particular host. It includes agents or software modules. The software of the HIDS has operated runs on the network machines like

routers, switches, and servers, etc. Thus, the agent of HIDS is operated like a consistent host. More information and details about HIDS can be found in [31,32].

- **Network-based IDS (NIDS):** This kind attempts to protect all machine systems in the IoT network. The architecture of NIDS (Network-based IDS). It comprises a group of sensors with particular functions located at a different location within the network. NIDS implementation can have a fantastic effect on the performance of the computer network. More information and details about HIDS can be found in [32,33].

2.2 Intrusion Detection Methodologies

Detection models are divided into two diverse types: signature or statistical-based models. The first type of signature model contrasts the traffic alongside a cluster of existing signatures. On the other side, the second type of statistical model preserves the users, hosts, applications, and connections profiles. Likewise, two key detection schemes are employed by the host or network: signature and anomaly-based models used to analyze the actions and discover intrusions. Consequently, there are three key intrusion detection methods, which are SD (Signature-based Detection), AD (Anomaly-based Detection), and SPA (Stateful Protocol Analysis). Tab. 1 demonstrates the prime disadvantages and advantages of the three detection methods. More information and details about the three key intrusion detection methods can be found in [31,34].

Table 1: Types of intrusion detection methodologies with their advantages and disadvantages

Intrusion detection system	Advantages	Disadvantages
Signature-based (knowledge-based)	• Detail contextual analysis. • An effective method to detect known attacks.	• Ineffective to detect unknown attacks, evasion attacks, and variants of known attacks. • Time-consuming to maintain the knowledge.
Anomaly-based (behavior-based)	• Effective to detect new vulnerabilities. • Less dependent on OS. • Facilitate detections of privilege abuse.	• Unavailable during the rebuilding of behavior profiles. • Difficult to trigger alerts at the right time.
Stateful protocol analysis (specification-based)	• Distinguish unexpected sequences of commands. • Know and trace the protocol states.	• Incapable of examining assaults which are like decent activities. • More consumption of resources for tracking protocol investigations.

2.3 Deep Learning

DL has been a pretty research subject in recent times for different domains [35]. DL depends on the unsupervised or supervised processes to discover classified depictions in deep structures [36]. The most aim of DL is to exploit the Artificial Neural Networks (ANN) to extract more levels of higher features. The categorization DL depends on the purpose and objective of structural design and

procedures. The division of DL algorithms can be categorized. More information and details about the DL models and structures can be found in [37,38].

2.4 Software Defined Network

The SDN (Software Defined Network) is an adaptable, manageable, dynamic, and valuable infrastructure [11]. It combines numerous network tools for dynamic and centralized computer network infrastructure management. It lets the administrator of a network manage the enterprise demands rapidly. The general architecture of the SDN consists of control, infrastructure, and application layers. It is noticed that the control and data planes are separated from the network devices' functionalities. More information and details about the SDN control and data planes can be found in [39,40].

The architecture of SDN extremely confirms network analysis and observing tools through the programming feature of the SDN controller. In [39], the authors suggested SDN-based IDS is presented in Fig. 1. This proposal aims to investigate the SDN network traffic for the malicious recognition process. The IDS is considered on the SDN controller identifies two procedures employed on the coming traffic from the switch to identify the intrusion attempts which decrease the performance of the SDN network infrastructure. Two main IDS techniques can be employed in SDN controllers: packet counter and time interval techniques. More information and details about the three key intrusion detection methods can be found in [39,40].

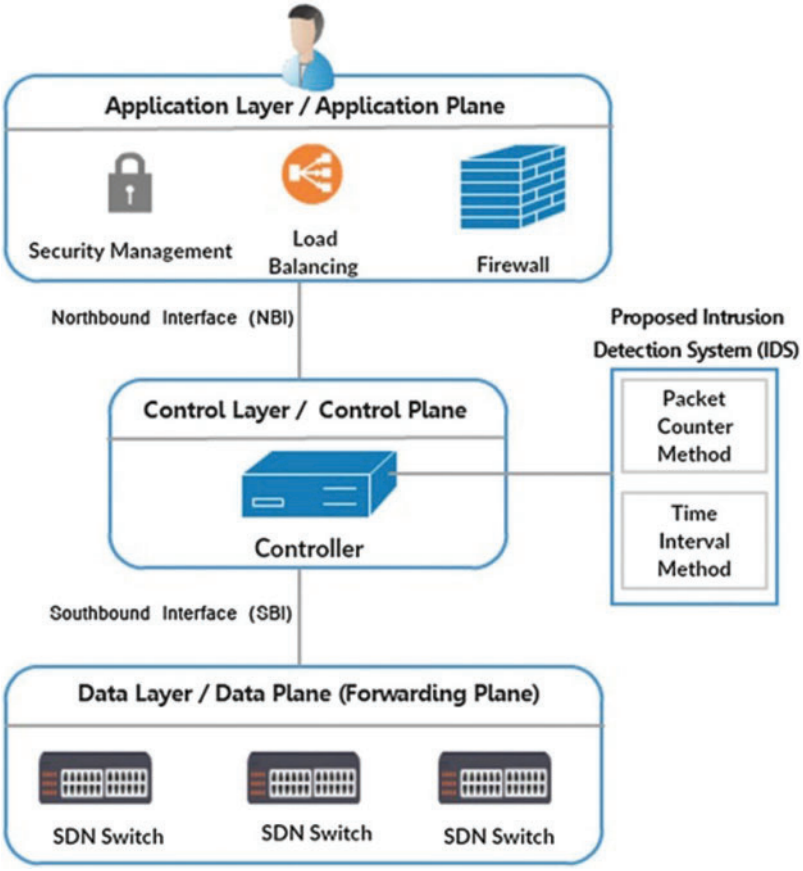

Figure 1: SDN-based intrusion detection system

3 Methodology and Implementation

Over recent years, some research works have been available on intrusion detection using data mining procedures like traffic data clustering and intrusion detection and classification. Portnoy [40] presented a new kind of clustering-based IDS scheme known as anomaly intrusion detection, which trains on unlabeled data to identify new intrusions. The authors in [41] suggested a new NSL-KDD data set, which composes a complete KDD data set.

Panda et al. [42] proposed a hybrid intelligent approach using a combination of classifiers to make the decision intelligently so that the overall performance of the resultant model is enhanced. Kang et al. [43] proposed an optimal feature selection algorithm that tackles the problem of choosing the optimal subset of features from several commonly used features to detect network intrusion that requires extensive computing resources. Xiu-yu [44] proposed a model of online attack detection for computer forensics to collect crime evidence of the attack. Siddiqui et al. [45] introduced an analysis of 10% of the KDD cup'99 training dataset based on intrusion detection. Also, they focused on establishing a relationship between the attack types and the protocol used by the hackers, using clustered data.

Subramanian et al. [46] presented an analysis of the effect of clustering the training data and test data in the classification efficiency of the Naive Bayes classifier. Kumar et al. [47] proposed a clustering approach based on a simple k-means clustering algorithm to analyze the NSL-KDD dataset. This proposed work provided a complete analysis of the NSL-KDD intrusion detection dataset. They tried to cluster the dataset into normal and four major attack categories such as DoS, Probe, R2L, U2R.

This paper proposes a deep learning (DL) model for building an efficient software-defined network (SDN)-based intrusion detection system. The NSL-KDD dataset is extensively analyzed and investigated using different clustering algorithms, including the proposed model. The utilized dataset is clustered into normal and four major attack categories. The results provide a complete analysis and a clear comparative study of different kinds of attacks included in the NSL-KDD dataset. Likewise, the results reveal that the proposed deep learning technique provides efficient intrusion detection performance compared to existing techniques.

4 Proposed Intrusion Data-Based Clustering and Detection Scenarios

This section offers a detailed explanation of the proposed work for traffic data clustering and intrusion detection in software-defined networks. The entire hybrid suggested system for SDN-based intrusion data clustering and classification is shown in Fig. 2.

4.1 Traffic Data-Based Clustering Algorithms

The clustering process is the data division into similar or dissimilar groups. The main benefit of the clustering process is the anomaly detection of intrusions deprived of preceding knowledge. This paper discusses the comparative study and investigation of five clustering algorithms: K-means, Farthest First, Canopy, Exception-maximization (EM), and Density-based algorithm. The description of the employed comparison clustering algorithms is introduced as follows.

1) **K-means clustering algorithm.** The K-means clustering algorithm is a cluster analysis method where we define K disjoint clusters based on the feature value of the objects to be grouped.
2) **Farthest first clustering algorithm.** It has the same procedure as the K-means algorithm; this also chooses centroids and assign the objects in the cluster but with max distance and initial seeds are the value which is at the largest distance to the mean of values, here cluster assignment

is a different clustering algorithm, at initial cluster, we get a link with high Session Count, like at cluster-0 more than in cluster-1, and so on.

3) **Canopy clustering algorithm.** It is a pre-clustering unsupervised scheme. It may be utilized as a pre-processing stage for the Hierarchical clustering scheme or the K-means scheme. It is a fast, simple, and accurate scheme in the clustering process, where each clustered object is arranged to be the only point within a multi-dimensional feature space. The canopy scheme utilizes two thresholds, T1 > T2 and the fast approximate distance metric for processing purposes.
4) **Exception-maximization (EM) clustering algorithm.** It is considered an expansion of the K-means scheme. It assigns the object to a cluster depending on a weight indicating the membership probability. So, there are no precise restrictions among different clusters. The EM scheme introduces a higher accuracy compared to the K-means scheme.
5) **Density-based clustering algorithm.** It is a data clustering algorithm. In the case of given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers point that lie alone in low-density regions (whose nearest neighbors are too far away).

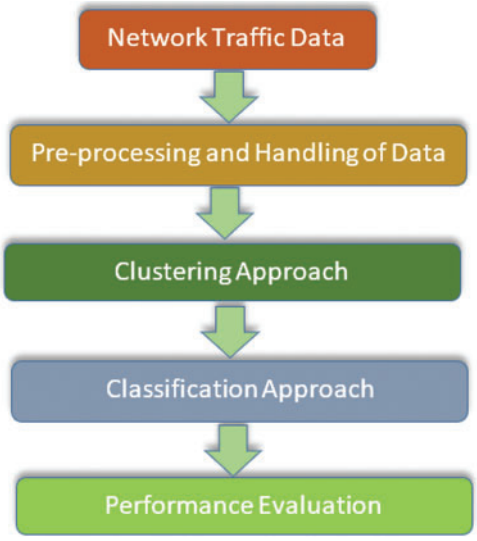

Figure 2: Proposed hybrid system combined clustering and classification for intrusion detection in SDN

4.2 SDN-Based IDS Using Deep Learning Model

This part introduces the proposed SDN-based IDS using deep learning. The proposed system can help to identify malicious attacks as intrusion actions. The proposed SDN-based deep learning model for the IDS process is given in Fig. 3. The dataset of NSL-KDD is utilized to evaluate the suggested SDN-based IDS using a DL scheme. All experiments have been performed using Python programming language on Spyder on Anaconda navigator software utilizing the Intel Core i5 GHz processor with 12 GB of RAM and 500 GB HD.

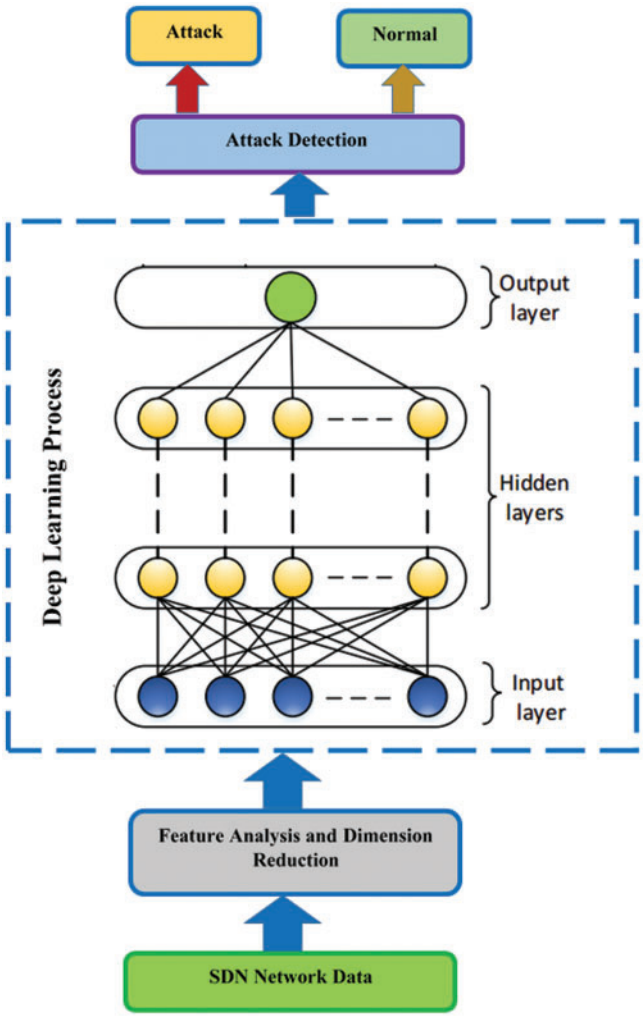

Figure 3: SDN-based deep learning model for intrusion detection

In this work, the NSL-KDD dataset is used to evaluate the results of applied algorithms. Dataset was introduced to resolve some intrinsic difficulties of the KDD-cup 1999 dataset [48,49]. The original KDD'99 dataset was composed of the test and train datasets that were utilized previously to evaluate the performance of IDS models. It combines three different categories of features: traffic-based, content-based, and basic features. The NSL-KDD dataset is a modern description of the KDD dataset. Therefore, it is employed in this paper to assess the feature selection subsets efficiency of the suggested schemes. Attacks in the dataset are categorized into four categories of U2R (User to Root) attack, R2L (Remote to Local) attack, probing attack, and DoS (Denial of Service) attack according to their characteristics as shown in Tab. 2.

Table 2: Attack categories in the NSL-KDD dataset

Attack category	Description
DoS	In the case of this attack, the service providers cannot permit legitimate users to utilize their services because these attacks consume network traffic and computing resource.
U2R	In the case of this attack, there are more attempts to access the root after acquiring access to ordinary users by sniffing and so on.
R2L	In the case of this attack, there are more attempts to access to the ordinary users by exploiting the vulnerability of the target server.
Probing	In the case of this attack, the attacker attempts to gather information about the network of computers for the apparent purpose of circumventing its security.

Therefore, the NSL-KDD dataset is utilized in this work to realize the evaluation and training of the suggested work. The NSL-KDD dataset has 41 attributes unfolding different traffic flow features, and a label is assigned either as a particular attack type or as normal data [50]. The features in the NSL-KDD dataset are of different data types. The testing dataset contains 38 attack types, while the training dataset includes 24 attack types. Tab. 3 presents the features with varying types of data in the NSL-KDD dataset.

Table 3: Features with various data kinds in the NSL-KDD dataset [51]

Data type	Feature
Nominal	2, 3, 4
Binary	7, 12, 14, 15, 21, 22
Numeric	1, 5, 6, 8, 9, 10, 11, 13, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41

5 Results and Discussions

This section illustrates the comparative results analysis, discussions and simulation environment setup of the proposed intrusion data-based clustering and detection scenarios.

5.1 Simulation Results of the Traffic Data-Based Clustering Algorithms Environment

The simulation results are performed with the WEKA environment using the NSL-KDD on a computer with a Core-i5 processor with 4 GB of RAM [52]. We performed a normalization process in the range of 0–1 to all input dataset attributes before the employment of the clustering process, and the number of clusters is set up to four.

The NSL-KDD dataset is analyzed using the K-means, Farthest First, Canopy, Exception-maximization (EM), and Density-based clustering algorithms, where the major attack types are presented in the training dataset. The performance of the aforementioned algorithms is evaluated based on the number of instances per cluster, the execution time, and incorrectly clustered instances.

Tabs. 4 to 8 and Figs. 4 to 8 depict the outcomes of the tested simulation results. Four clusters are distributed for the clustered instances, clusters 0 to 3, including normal cases. The clustered instances distribution is classified into Normal, DoS, R2L, U2R, and Probe.

Table 4: The simulation results of the K-means clustering algorithm

Cluster number	No. of instances	Percentage	Normal	dos	probe	r2l	u2r
Cluster 0	37026	25%	101	36109	807	9	0
Cluster 1	41533	28%	35166	734	4328	1233	72
Cluster 2	49724	34%	38820	6216	2182	2459	47
Cluster 3	19624	13%	2880	9928	6637	179	0
Time is taken to build a model (full training data): 4.51 s.							
Incorrectly clustered instances: 65108 with a Percentage of 44.02%.							

Table 5: The simulation results of the farthest first clustering algorithm

Cluster number	No. of instances	Percentage	Normal	dos	probe	r2l	u2r
Cluster 0	51452	35%	11092	38749	1511	90	10
Cluster 1	30342	21%	9789	12280	7654	619	0
Cluster 2	59970	41%	54145	1671	1092	2955	107
Cluster 3	6143	4%	1941	287	3697	216	2
Time is taken to build a model (full training data): 0.39 s.							
Incorrectly clustered instances: 47143 with Percentage of 31.87%.							

Table 6: The simulation results of the canopy clustering algorithm

Cluster number	No. of instance	Percentage	Normal	dos	probe	r2l	u2r
Cluster 0	67355	46%	57738	2546	4001	2962	108
Cluster 1	37114	25%	135	36162	808	9	0
Cluster 2	23934	16%	16357	4344	2494	728	11
Cluster 3	19504	13%	2737	9935	6651	181	0
Time is taken to build the model (full training data): 2.53 s.							
Incorrectly clustered instances: 46628 with a Percentage of 31.53%.							

Table 7: The simulation results of the EM clustering algorithm

Cluster number	No. of instance	Percentage	Normal	dos	probe	r2l	u2r
Cluster 0	10157	7%	6122	2478	284	1189	84
Cluster 1	20481	14%	3206	10404	6705	166	0
Cluster 2	36302	25%	7	35714	580	1	0
Cluster 3	80967	55%	67632	4391	6385	2524	35
Time is taken to build a model (full training data): 40.48 s.							
Incorrectly clustered instances: 36667 with a Percentage of 24.79%.							

Table 8: The simulation results of the density-based clustering algorithm

Cluster number	No. of instance	Percentage	normal	dos	probe	r2l	u2r
Cluster 0	36668	25%	45	36050	571	2	0
Cluster 1	62276	42%	55038	1050	4450	1655	83
Cluster 2	28858	20%	18858	5620	2321	2023	36
Cluster 3	20105	14%	3026	10267	6612	200	0
Time is taken to build a model (full training data): 5.41 s.							
Incorrectly clustered instances: 48184 with a Percentage of 32.58%.							

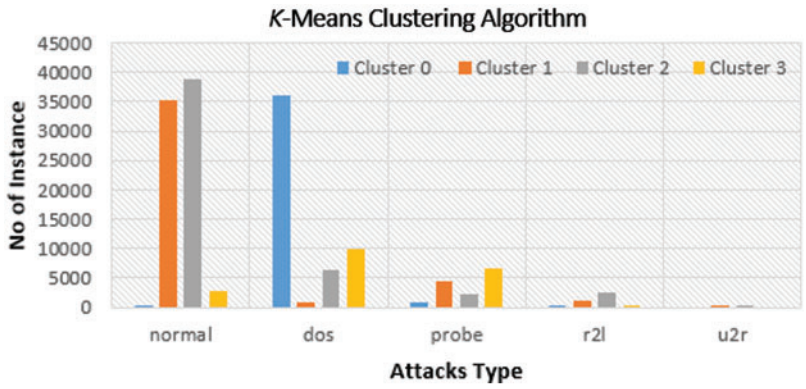

Figure 4: Distribution of instances to clusters using the K-means clustering algorithm

Fig. 4 gives the clustered instances results using the K-means algorithm. The number of clustered instances in the case of each tested cluster is presented in Tab. 4. Moreover, this table presents the distribution of the instances of each attack. It is noticed that the K-means scheme takes 4.51 s to build cluster models and the incorrectly clustered instances are 65108. Fig. 5 shows the clustered instances results using the Farthest First algorithm. The number of clustered instances in the case of each tested

cluster is introduced in Tab. 5. Furthermore, this table presents the distribution of the instances of each attack. It is observed that the Farthest First scheme takes 0.39 s to build cluster models and the incorrectly clustered instances are 47143.

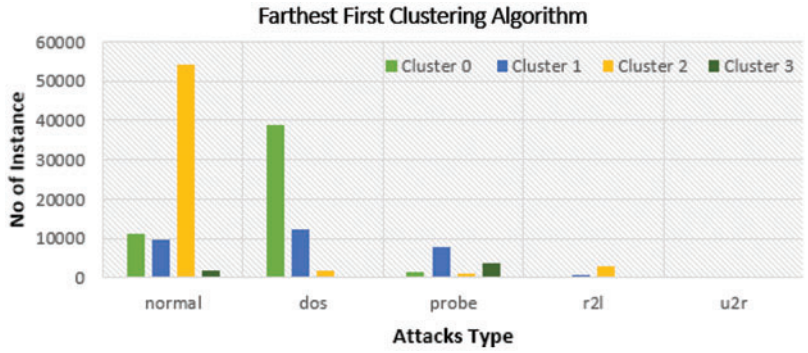

Figure 5: Distribution of instances to clusters using the farthest first clustering algorithm

Fig. 6 presents the clustered instances results using the Canopy algorithm. The number of clustered instances in the case of each tested cluster is given in Tab. 6. Moreover, this table presents the distribution of the instances of each attack. It is noticed that the Canopy scheme takes 2.53 s to build cluster models and the incorrectly clustered instances are 46628. Fig. 7 shows the clustered instances results using the Exception-maximization (EM) algorithm. The number of clustered instances in the case of each tested cluster is presented in Tab. 7. Furthermore, this table presents the distribution of the instances of each attack. It is observed that the EM scheme takes 40.48 s to build cluster models and the incorrectly clustered instances are 36667.

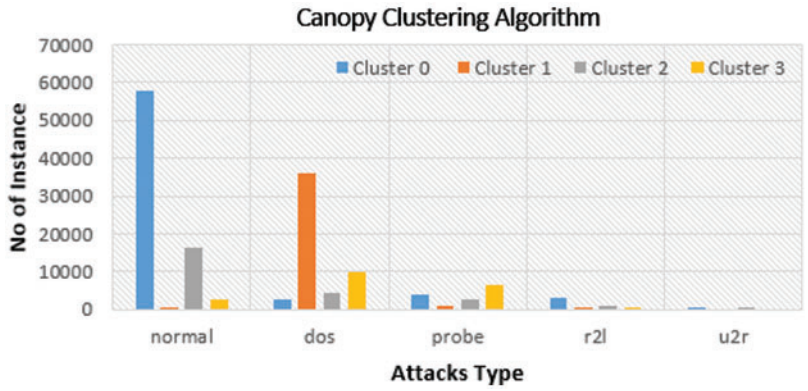

Figure 6: Distribution of instances to clusters using the canopy clustering algorithm

Fig. 8 shows the clustered instances results using the Density-based clustering algorithm. The number of clustered instances in the case of each tested cluster is introduced in Tab. 8. Moreover, this table presents the distribution of the instances of each attack. It is noticed that the Density-based clustering scheme takes 5.41 s to build cluster models and the incorrectly clustered instances are 48184. Tab. 9 and Fig. 9 show a comparison between the five clustering algorithms based on the number of instances between the four clusters. Tab. 10 and Fig. 10 show a comparison between the five algorithms based on the execution time. Tab. 11 and Fig. 11 show a comparison between five clustering algorithms based on the number of incorrectly clustered instances.

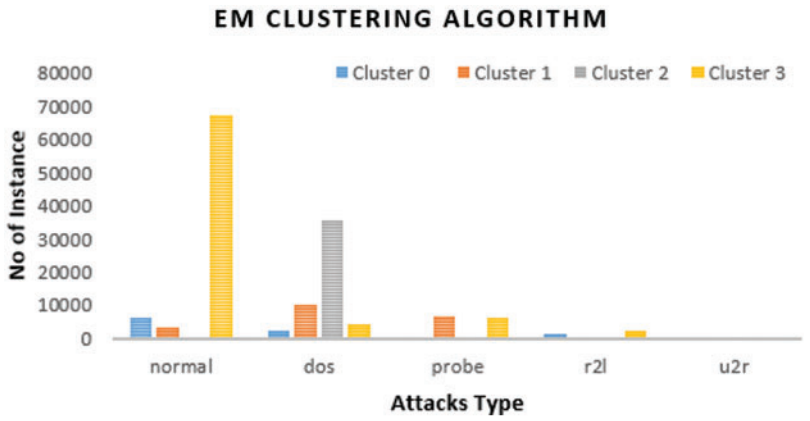

Figure 7: Distribution of instances to clusters using the EM clustering algorithm

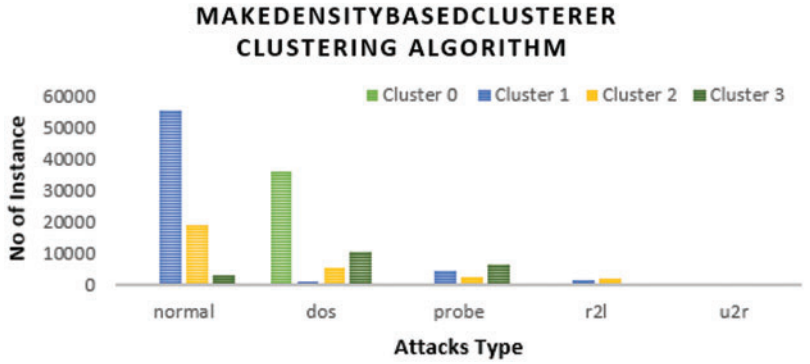

Figure 8: Distribution of instances to clusters using the density-based clustering algorithm

From all the presented simulation and comparison results, it is obvious that the distribution of instances is various from one cluster to another. It is also noticed that the Farthest First clustering algorithm takes less execution time among the five clustering algorithms. Moreover, the EM clustering algorithm gives a great number of incorrectly clustered instances among the five algorithms.

Table 9: The comparison results between five clustering algorithms

Cluster number	K-means	Farthest first	Canopy	EM	Density-based
Cluster 0	37026	51452	67355	10157	36668
Cluster 1	41533	30342	37114	20481	62276
Cluster 2	49724	59970	23934	36302	28858
Cluster 3	19624	6143	19504	80967	20105

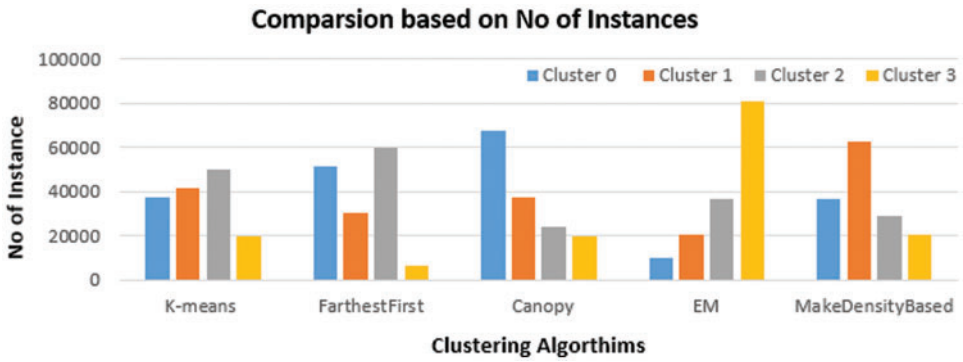

Figure 9: Comparison between clustering algorithms based many of instances

Table 10: Comparison between the clustering algorithms based on execution time

Algorithm	Time (s)
K-means	4.51
Farthest first	0.39
Canopy	2.53
EM	40.48
Density-based	5.41

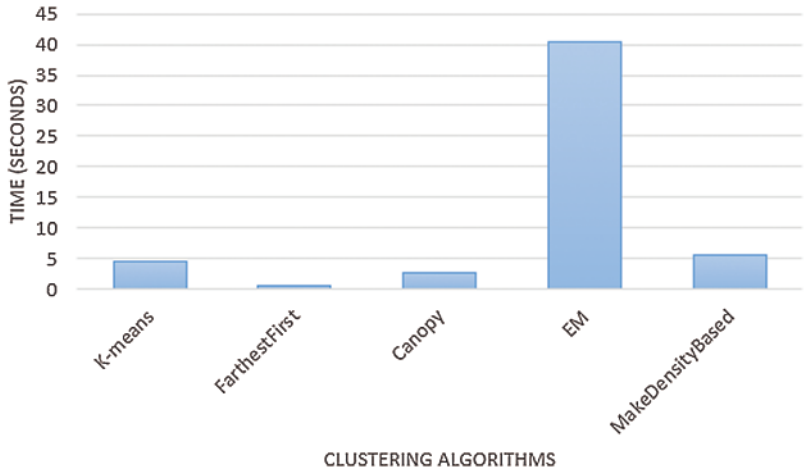

Figure 10: Comparison of results between the clustering algorithms based on execution time

Table 11: Comparison of between the clustering algorithms based on the incorrectly clustered instances

Algorithm	Incorrectly clustered instances	
K-means	65108	44.0196%
Farthest first	47143	31.8734%
Canopy	46628	31.5252%
EM	36667	24.7906%
Density-based	48184	32.5772%

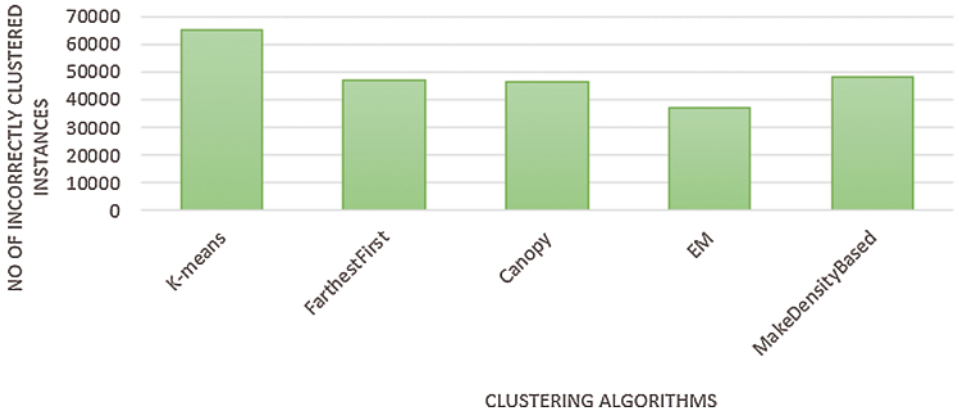

Figure 11: Comparison between the five clustering algorithms based on the number of incorrectly clustered instances

5.2 Simulation Results of the SDN-Based IDS Using Deep Learning Model

The deep learning-based IDS model is implemented for binary class attack classification. It classifies the input data whether it belongs to the normal or attack. The classification accuracy is evaluated for all 41 features, where the accuracy of detection of different attack classes was higher using the proposed deep learning model compared to other machine learning techniques such as LR, LDA, and NB as tabulated in Tab. 12 and Fig. 12.

Table 12: Accuracy based comparison of different algorithm

Algorithm	Accuracy
Logistic Regression (LR)	15.049%
Linear Discriminant Analysis (LDA)	82.333%
Gaussian Naive Bayes (NB)	51.889%
Proposed Deep Learning (DL)	94.21%

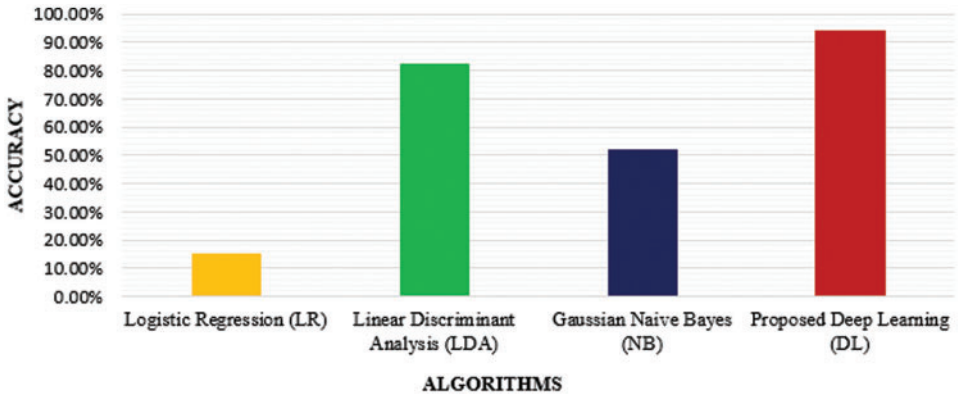

Figure 12: Comparison of the proposed DL algorithm with different algorithms

In the suggested model, the training percentage is 20%, epochs are 200, the batch size is 10, and the number of selected attributes is 16. The accuracy and loss percentage with different types of epochs are shown in Figs. 13 and 14, wherewith increasing the number of epochs, the accuracy percentage is increased, and the loss percentage is decreased.

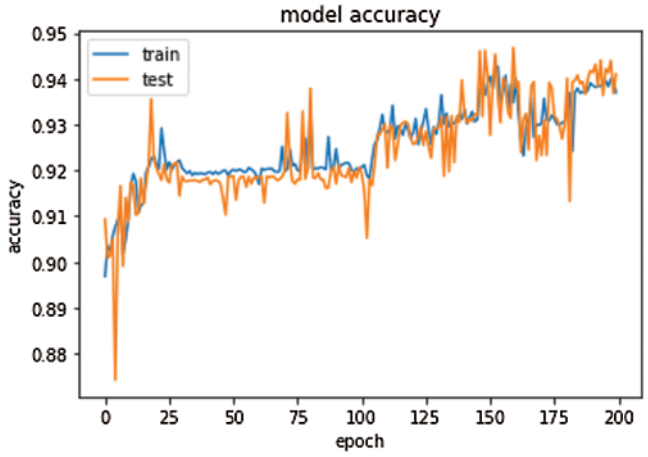

Figure 13: Proposed deep learning model accuracy

5.3 Comparative Analysis

From Tab. 13, the importance of the proposed scheme is quite evident. The proposed approach is an attempt to build an efficient software-defined network (SDN)-based intrusion detection system using deep learning. Besides, extensively analyzing and investigating the NSL-KDD dataset using different clustering algorithms into normal and four major attack categories. The results conceal that the proposed deep learning technique provides efficient intrusion detection performance compared to existing standard techniques. The results deliver a broad analysis and a rich comparative study of different kinds of attacks in this dataset.

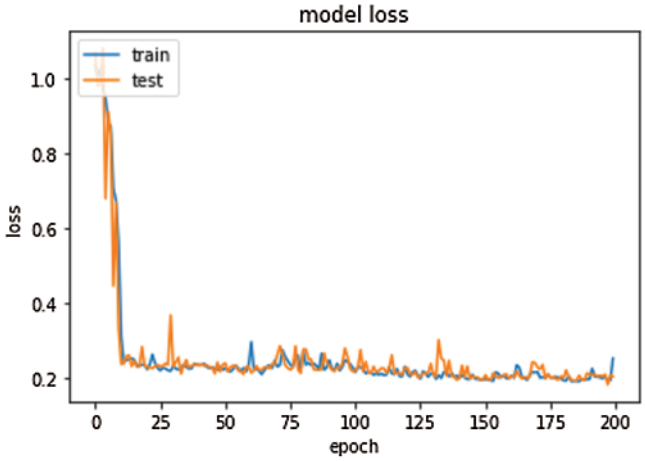

Figure 14: Proposed deep learning model loss

Table 13: Comparative analysis with the previously proposed systems

Scheme	Year	Task	Infrastructure	Dataset	Technique
[1]	2020	Detection	Wireless networks	NSL-KDD	DLSTM
[2]	2019	Detection	IoT	Bot-IoT	SVM
[3]	2018	Detection	Cloud	CICIDS2017, NSL-KDD version 2015 and CIDDS-001	Deep neural network
[4]	2019	Detection	Computer networks	CICIDS2017	MLP and payload classifier
Proposed approach	**2022**	**Clustering and detection**	**SDN**	**NSL-KDD**	**Deep neural network**

6 Conclusion and Future Work

In recent times, recognizing and detecting anonymous threats and risks is a significant task in providing a secure SND system by applying an efficient intrusion detection system. Therefore, this paper provides a clustering-based analysis of the NSL-KDD dataset using the K-means, Farthest First, Canopy, Exception-maximization (EM), and Density-based clustering algorithms. Likewise, it presents a deep learning system for building an efficient intrusion detection system to detect unknown malicious and illegitimate activities. The simulation results show that using the Farthest First scheme introduced superior instances distribution compared to the other four clustering schemes. The deep learning model also provides high-performance accuracy compared with existing algorithms in detecting intrusion events. For example, the proposed DL method achieved a high detection accuracy of 94.21%. In our future work, we intend to employ advanced artificial intelligence tools to detect the new versions of attacks in the SDN network. In addition, the real implementation of an IoT-based SDN network in cybersecurity applications will be developed and designed.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Q. Yan, F. Yu, Q. Gong and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2015.

[2] J. Ali, B. Roh, B. Lee, J. Oh and M. Adil, "A machine learning framework for prevention of software-defined networking controller from DDoS attacks and dimensionality reduction of big data," in *Proc. IEEE Int. Conf. on Information and Communication Technology Convergence (ICTC)*, Jeju, South Korea, pp. 515–519, 2020.

[3] J. Ali and B. Roh, "An effective hierarchical control plane for software-defined networks leveraging TOPSIS for end-to-end QoS class-mapping," *IEEE Access*, vol. 8, pp. 88990–89006, 2020.

[4] K. Bakshi, "Considerations for software defined networking (SDN): Approaches and use cases," in *Proc. IEEE Aerospace Conf.*, Big Sky, MT, USA, pp. 1–9, 2013.

[5] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Computer Networks*, vol. 112, no. 3, pp. 279–293, 2017.

[6] R. Khondoker, A. Zaalouk, R. Marx and K. Bayarou, "Feature-based comparison and selection of software defined networking (SDN) controllers," in *Proc. World Congress on Computer Applications and Information Systems (WCCAIS)*, Hammamet, Tunisia, pp. 1–7, 2014.

[7] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *Journal of Network and Computer Applications*, vol. 80, no. 4, pp. 200–218, 2017.

[8] H. Kwon, "Defending deep neural networks against backdoor attack by using de-trigger autoencoder," *IEEE Access*, vol. 9, pp. 2169–3536, 2021.

[9] H. Kwon and J. Baek, "Adv-plate attack: Adversarially perturbed plate for license plate recognition system," *Journal of Sensors*, vol. 5, pp. 1–16, 2021.

[10] T. Tang, L. Mhamdi, D. McLernon, S. Zaidi and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, pp. 258–263, 2016.

[11] 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, 2018.

[12] T. Tang, L. Mhamdi, D. McLernon, S. Zaidi and M. Ghogho, "Deep recurrent neural network for intrusion detection in SDN-based networks," in *Proc. 4th IEEE Conf. on Network Softwarization and Workshops (NetSoft)*, Montreal, QC, Canada, pp. 202–206, 2018.

[13] X. Huang, T. Yuan, G. Qiao and Y. Ren, "Deep reinforcement learning for multimedia traffic control in software defined networking," *IEEE Network*, vol. 32, no. 6, pp. 35–41, 2018.

[14] E. Hemdan and D. Manjaiah, "Digital investigation of cybercrimes based on big data analytics using deep learning," in *Proc. Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications*, 1st edition, vol. 2, USA: IGI Global, pp. 615–632, 2020.

[15] J. Jabez and B. Muthukumar, "Intrusion detection system (IDS): Anomaly detection using outlier detection approach," *Procedia Computer Science*, vol. 48, no. 7, pp. 338–346, 2015.

[16] S. Duque and M. Omar, "Using data mining algorithms for developing a model for intrusion detection system (IDS)," *Procedia Computer Science*, vol. 61, pp. 46–51, 2015.

[17] C. Gonzalez, S. M. Charfadine, O. Flauzac and F. Nolot, "SDN-based security framework for the IoT in distributed grid," in *Proc. Int. Multidisciplinary Conf. on Computer and Energy Science (IMCCES)*, Split, Croatia, pp. 1–5, 2016.

[18] Y. Liu, Y. Kuang, Y. Xiao and G. Xu, "SDN-based data transfer security for internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 257–268, 2017.

[19] W. El-Shafai and E. Hemdan, "Robust and efficient multi-level security framework for color medical images in telehealthcare services," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 2, pp. 1–16, 2021.
[20] O. Al-Jarrah and A. Arafat, "Network intrusion detection system using attack behavior classification," in *Proc. IEEE Int. Conf. on Information and Communication Systems (ICICS)*, Irbid, Jordan, pp. 1–6, 2014.
[21] M. Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems," *Computer Communications*, vol. 34, no. 1, pp. 107–117, 2011.
[22] A. Alarifi, S. Sankar, T. Altameem, K. Jithin and W. El-Shafai, "A novel hybrid cryptosystem for secure streaming of high efficiency H. 265 compressed videos in IoT multimedia applications," *IEEE Access*, vol. 8, pp. 128548–128573, 2020.
[23] P. Sangkatsanee, N. Wattanapongsakorn and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Computer Communications*, vol. 34, no. 18, pp. 2227–2235, 2011.
[24] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.
[25] S. Ibrahim, M. Egila, H. Shawky, M. Elsaid and W. El-Shafai, "Cancelable face and fingerprint recognition based on the 3D jigsaw transform and optical encryption," *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 14053–14078, 2020.
[26] B. Yang, X. Fu, N. Sidiropoulos and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. Int. Conf. on Machine Learning*, Sydney, Australia, pp. 3861–3870, 2017.
[27] W. Lin, S. Ke and C. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, no. 4, pp. 13–21, 2015.
[28] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.
[29] S. Otoum, B. Kantarci and H. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Networking Letters*, vol. 1, no. 2, pp. 68–71, 2019.
[30] U. Shafique and H. Qaiser, "A comparative study of data mining process models (KDD, CRISP-DM and SEMMA)," *International Journal of Innovation and Scientific Research*, vol. 12, no. 1, pp. 217–222, 2014.
[31] S. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing*, vol. 10, no. 1, pp. 1–35, 2010.
[32] I. Raghav, S. Chhikara and N. Hasteer, "Intrusion detection and prevention in cloud environment: A systematic review," *International Journal of Computer Applications*, vol. 68, no. 24, pp. 7–11, 2013.
[33] D. Mudzingwa and R. Agrawal, "A study of methodologies used in intrusion detection and prevention systems (IDPS)," in *Proc. IEEE Southeastcon*, Orlando, USA, pp. 1–6, 2012.
[34] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 4, pp. 197–387, 2014.
[35] W. El-Shafai, S. El-Rabaie, M. El-Halawany and F. Abd El-Samie, "Efficient hybrid watermarking schemes for robust and secure 3D-MVC communication," *International Journal of Communication Systems*, vol. 31, no. 4, pp. 1–18, 2018.
[36] N. Soliman, M. Khalil, A. Algarni, S. Ismail and W. El-Shafai, "Efficient HEVC steganography approach based on audio compression and encryption in QFFT domain for secure multimedia communication," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4789–4823, 2021.
[37] Y. Lin, H. Lei, X. Li and J. Wu, "Deep learning in NLP: Methods and applications," *Journal of University of Electronic Science and Technology of China*, vol. 46, no. 6, pp. 913–919, 2017.
[38] Y. Hande, A. Muddana and S. Darade, "Software-defined network-based intrusion detection system," in *Proc. Innovations in Electronics and Communication Engineering*, Springer, Lecture Notes in Networks and Systems, vol. 7, Singapore: Springer, pp. 535–543, 2017.
[39] P. Wang, K. Chao, H. Lin, W. Lin and C. Lo, "An efficient flow control approach for SDN-based network threat detection and migration using support vector machine," in *Proc. IEEE 13th Int. Conf. on e-Business Engineering (ICEBE)*, Macau, China, pp. 56–63, 2016.

[40] W. El-Shafai, "Joint adaptive pre-processing resilience and post-processing concealment schemes for 3D video transmission," *3D Research*, vol. 6, no. 1, pp. 1–10, 2015.
[41] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, pp. 1–6, 2009.
[42] M. Panda, A. Abraham and M. Patra, "A hybrid intelligent approach for network intrusion detection," *Procedia Engineering*, vol. 30, no. 8, pp. 1–9, 2012.
[43] S. Kang and K. Kim, "A feature selection approach to find optimal feature subsets for the network intrusion detection system," *Cluster Computing*, vol. 19, no. 1, pp. 325–333, 2016.
[44] Z. Xiu-yu, "A model of online attack detection for computer forensics," in *Proc. Int. Conf. on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, China, vol. 8, pp. 533–537, 2010.
[45] M. Siddiqui and S. Naahid, "Analysis of KDD CUP 99 dataset using clustering-based data mining," *International Journal of Database Theory and Application*, vol. 6, no. 5, pp. 23–34, 2013.
[46] U. Subramanian and H. Ong, "Analysis of the effect of clustering the training data in naive Bayes classifier for anomaly network intrusion detection," *Journal of Advances in Computer Networks*, vol. 2, no. 1, pp. 85–88, 2014.
[47] V. Kumar, H. Chauhan and D. Panwar, "K-means clustering approach to analyze NSL-KDD intrusion detection dataset," *International Journal of Soft Computing and Engineering*, vol. 3, no. 4, pp. 1–4, 2013.
[48] KDD Cup 1999 Data. 2019. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/, last access on 2-12-2019.
[49] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.
[50] S. Potluri and C. Diedrich, "Deep feature extraction for multi-class intrusion detection in industrial control systems," *International Journal of Computer Theory and Engineering*, vol. 9, no. 5, pp. 374–379, 2017.
[51] W. El-Shafai, E. El-Rabaie, M. El-Halawany and F. Abd El-Samie, "Efficient multi-level security for robust 3D color-plus-depth HEVC," *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 30911–30937, 2018.
[52] R. Sadoddin and A. Ghorbani, "A comparative study of unsupervised machine learning and data mining techniques for intrusion detection," in *Proc. Int. Workshop on Machine Learning and Data Mining in Pattern Recognition*, Leipzig, Germany, pp. 404–418, 2007.