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

Mrs. M. Sandhya Vani[1] ,Mrs. R.Durga Devi[2], Dr. Deena Babu Mandru[3]

[1][2] Assistant Professor, Department of Information Technology, Malla Reddy Engineering College, Hyderabad, Telangana
[3] Professor, Department of Information Technology, Malla Reddy Engineering College, Hyderabad, Telangana

Abstract: The board of directors and mixed media mining strategies are excited about further research and development of the organization's traffic processes. Relying on a unified, programmable controller, security has recently become the most complex task in Software Defined Networking (SDN). Next, observing the organization's traffic is essential to distinguish and discover outage anomalies in the SDN environment. Therefore, this paper provides an in-depth review and examination of the NSL-KDD dataset using five different clustering calculations: K-implies, further before, canopy, density-based computation, and expansion of exceptions (EM), contrasting these five calculations largely involving Waikato Environment Programming for Knowledge Analysis (WEKA). Additionally, this paper introduces an SDN-based outage identification framework that uses a deep learning (DL) model and the Knowledge Discovery in Databases (KDD) dataset. The dataset used is first grouped into typical and four major attack classifications using the grouping s ystem. Next, a deep learning technique is proposed to promote a successful SDN-based outage detection framework. The findings provide in-depth investigation and flawless intelligent investigation into the different types of attacks recorded for the KDD dataset. Furthermore, the results show that the proposed deep learning strategy outperforms existing methods with regards to interrupt discovery performance. For the analyzed dataset, for example, the proposed technique achieves an identification accuracy of 94.21%.

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

1 Introduction

SDN was recently created as one of the competent responses to change the ultimate fate of global organizations and the Internet [1-3]. SDN, another emerging innovation, isolates information and control planes. The control plan is equipped to address all safety issues in the organization [4-7].

Deep learning (DL) is a kind of model of artificial intelligence, as it depends on the cycle of extracting huge information with confusing structures [8,9]. It is useful when gaining a lot of information unaided [10,11]. Using deep learning produces some benefits, including order execution and the quality of more developed tests. These benefits can lead to traditional AI applications, such as normal programmed language handling, speech recognition, and others.

The Intrusion Detection System (IDS) can be used to monitor network traffic, unauthorized businesses and organization [15,16]. An investigation engine, sensors and a revealing panel are important to the structure of the IDS. SDN-based security is a basic model for controlling malicious flows in SDN switches [17,18]. As a result, with the growing number of attacks and severe threats against a variety of computer systems such as a computer, organization, cloud and object network, the investigation of interrupt detection systems (IDS) has received a lot of attention. by some security specialists in recent years [19]. The IDS model can identify malicious attacks such as data theft [20,21]. The National Institute of Standards and Technology (NIST) [22] describes the invasion or disruption as "a job to organize the CIA .

Break location using untagged information, also called grouping, can gather related records into similar groups. Then, using the normal distance measurements on these clusters, the peculiarities can be solved [26]. Clustering is a kind of unaided discovery that works with unlabeled information [27]. Artificial intelligence-based disruption location (ML) examination may support the discovery of strange behaviors in the organization [28]. Regardless, there are some break ID calculations available, but your exhibit needs to move on. DL is a subset of ML strategies. Deep learning has gained notoriety as an AI point of view.

Therefore, DL strategies were used to distinguish some sample types or lists. Furthermore, the use of deep learning can help work on the accuracy and execution of IDS systems [29]. To carry out and understand the information search for this work, there are some information mining process models such as attachment [30]:

KDD (Knowledge Discovery Databases): KDD is a cycle by which specialists separate examples and experiences from information. It is divided into five phases: translation / evaluation, information extraction, choice of changes and pre-processing. This model is used in this revision to end the document's responsibility to create an interrupt detection framework.
SEMMA (Sampling, Exploring, Modifying, Modeling, and Accessing) - Although the SEMMA model is similar in design to the KDD, it is easier to apply to general information examination tasks as it does not focus so deeply on the specific phases of the learning. information.
Fresh DM (Cross-Industry Standard Process for Data Mining): This model, recently developed by IBM for data mining activities, is useful for virtually all data analysis projects.
This exam paper explores, analyzes, and investigates five grouping calculations: K-Implies, Farthest First, Canopy, Density-based Calculus, and Exception-boost (EM), using WEKA programming to thoroughly analyze these five calculations. The essential goal of this work is to study the NSL-KDD dataset. Additionally, this document presents an SDN-based interrupt location framework that uses a DL model with the NSL-KDD dataset.

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 Outage Identification Framework (IDS) is a product or equipment framework that analyzes network exercises or frameworks for forms of malicious behavior and creates reports for executives and regulatory frameworks. The essential ability of interrupt detection and contrast (IPS / IDS) frameworks is to distinguish attacks and interrupts. In addition, they are used to prevent people and display log alerts from bypassing security systems, as well as to recognize problems with security rules. The Interrupt Discovery and Anticipation Framework (IDS / IPS) has become an underlying security concern. For the interrupt detection process, a variety of procedures [31] can be used. They are used to protect PC networks from a variety of attacks. However, you can also dispose of the textures or terminate the association. The following are the characterizations of the interrupt frames [32,33].
Have Based Interrupt Discovery Frameworks (HIDS): HIDS is used to protect a particular host. It consists of specialists or programming modules. HIDS programming was performed on network machines, for example, Switches, switches and servers, for example. Thereafter, the HIDS specialist acts as a trusted host. [31,32] provides additional data and subtleties on HIDS.
Network-Based Interrupt Location Framework (NIDS) - This type attempts to defend all machine frameworks in the IoT organization. The NIDS (network-based IDS) design. It consists of a set of sensors with explicit capabilities that are appropriate throughout the organization. Running NIDS can greatly affect your PC's organization view. [32,33] provides additional data and subtleties on HIDS.

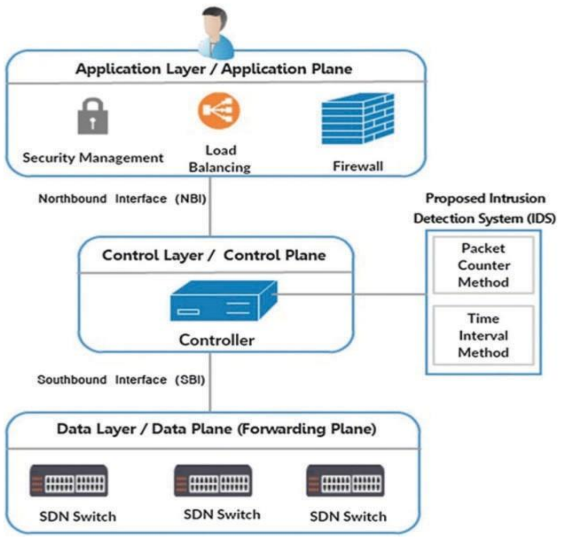
2.2 Intrusion Detection Methods

Recognition models are organized into two types: signature-based models and fact-based models. The main type of brand model analyzes traffic to a collection of existing brands. The second type of factual model, on the other hand, maintains the profiles of clients, hosts, applications, and associations. In addition, the host or organization uses two key discovery plans: flagging and irregularity-based models, which are used to analyze activities and distinguish outages. Next, there are three key interrupt placement strategies: Signature-Based Detection (SD), Anomaly-Based Detection (AD), and Stateful Protocol Analysis (SPA). Tab. 1 shows the essential disadvantages and advantages of the three localization techniques. [31,34] provides further information and subtleties on the three main techniques for locating the interruption.

2.3 Software Defined Network

The Software Defined Network (SDN) framework is versatile, sensible, dynamic and important [11]. Consolidate multiple staging devices for dynamic and integrated management of the PC network base. It allows network managers to quickly monitor risky requests. SDN engineering consists of three layers: control, base and application. In fact, it is quite significant that

information and control plans are specific to the functionalities of organizational devices. [39,40] provides further insights and nuances on SDN information and control plans.
Through the programming element of the SDN controller, SDN engineering incredibly affirms network search and observation tools. The creators of [39] proposed an SDN-based IDS, shown in Fig. 1. This proposal seeks to investigate SDN network traffic for malicious recognition. The Interrupt Discovery Structure (IDS) in the SDN controller distinguishes two systems used in inbound switch traffic to recognize interrupt attempts that reduce the core exposure of the SDN. Two main IDS strategies can be used in SDN controllers: packet counter and time extension procedures. [39,40] provide additional data and subtleties on the three main interrupt acknowledgment techniques.


Application Layer / Application Plane
Security Management
Load Balancing
Firewall
Northbound Interface (NBI)
Proposed Intrusion Detection System (IDS)
Control Layer / Control Plane
Controller
Packet Counter Method
Time Interval Method
Southbound Interface (SBI)
Data Layer / Data Plane (Forwarding Plane)
SDN Switch
SDN Switch
SDN Switch
Fig 1 : Intrusion detection system based on SDN

1 Methodology and Implementation

In recent years, some research on interrupt discovery has become accessible using data mining strategies such as grouping of traffic data and interrupt acknowledgment and sorting. Portnoy [40] introduced another type of cluster-based IDS graph known as inconsistency break detection, which is trained on unlabeled information to distinguish new interrupts. The creators in [41] proposed another collection of NSL-KDD information, which indexes the total KDD information.

Some scans involve interrupt detection using data mining strategies, such as clustering of traffic information and interrupt acknowledgment and sorting have recently been opened. Portnoy [40] introduced abnormal interrupt identification, another type of combined IDS diagram that is trained on unlabeled information to identify new interrupts. The creators of [41] have proposed another NSL-KDD information index that compiles a total information KDD index. This paper proposes a deep learning (DL) model to promote a competent SDN-based interrupt detection framework. The NSL-KDD dataset is inspected and examined in its entirety using several clustering calculations, including the proposed model. The dataset used is isolated into ordinary classifications and four major assaults. The findings provide a long-range investigation and an unambiguous close investigation into the different types of attacks remembered by the NSL-KDD dataset.

4. Proposed Intrusion Data-Based Clustering and Detection Scenarios

This segment provides a definitive clarification of the proposed work for the aggregation of traffic information and the location of interruptions in the schedule of the characterized networks. The whole half and half recommended framework for clustering and characterization of SDN based outage information is shown in Fig. 2.

4.1 Clustering algorithms based on traffic data

The grouping system divides information into comparative and disparate sets. The main advantage of the clustering system is the identification of rarities in outages without prior information. This paper examines and discusses five clustering calculations: K-implies, further before, canopy, expansion of exceptions (EM), and density-based computation. Below is a description of the correlation used to group the calculations.

i. The calculation of the K-cluster implies. The calculation of the K-implicate cluster is a group search strategy in which disjoint K clusters are characterized based on the value of the elements to be assembled.

ii. The calculation of the first pool furthest away. A strategy similar to that of calculating the K-implies follows; choosing centroids and assigning objects to groups, however, with extreme distance and initial seeds which are the closest qualities to the mean of the values; the group activity is an alternative grouping calculation; in the pool below, we get a connection with a large number of sessions, for example in pool 0 a greater number than in pool 1, etc.

iii. Algorithm for the grouping of tones. It's a solo pre-group plan. It tends to be used as a pre-management stage for hierarchical clustering or K-imply plans. It is a quick, basic, and exact grouping plan in which each grouped object is arranged to be the main point in a complex element space. For management, the shelter plot uses two limits, T1> T2, and the fast approximate distance metric.

iv. The Exception-boost (EM) clustering computation is seen as an expansion of the K-imply conspiracy. Distribute the object to a group in light of a weight that shows the probability of participation. Consequently, there are no exact limits between the various clusters. Compared to the K-imply storyline, the EM conspiracy offers more remarkable accuracy.

v. Grouping algorithm based on thickness. It is a grouping calculation for information. When assigned multiple fires in a space, it picks up fires that are very close together (fires with many close neighbors), recognizing disconnected fires in thin areas (whose closest neighbors are too far apart) as anomalies.

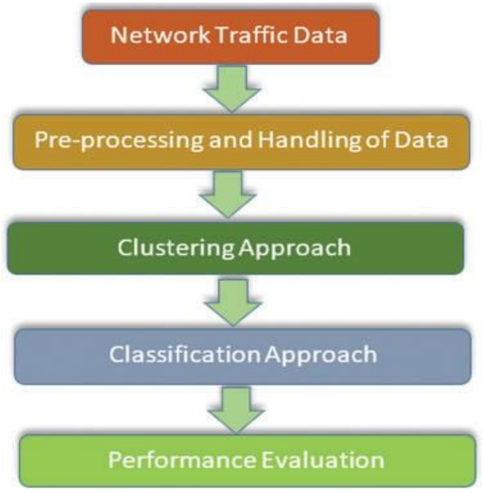


Figure 2: Proposed hybrid system combined clustering and classification for intrusion detection in SDN

SDN-Based IDS Using Deep Learning Model

This segment depicts the proposed SDN-put together IDS based with respect to profound learning. The proposed framework can aid the ID of pernicious assaults as interruption activities. Figure 3 portrays the proposed SDN-based profound learning model for the IDS interaction. The NSL-KDD dataset is utilized to assess the proposed SDN-based IDS utilizing a DL plot. All tests were completed on Spyder utilizing the Python programming language and the Anaconda guide programming, with an Intel Core i5 GHz processor, 12 GB of RAM, and a 500 GB hard drive..

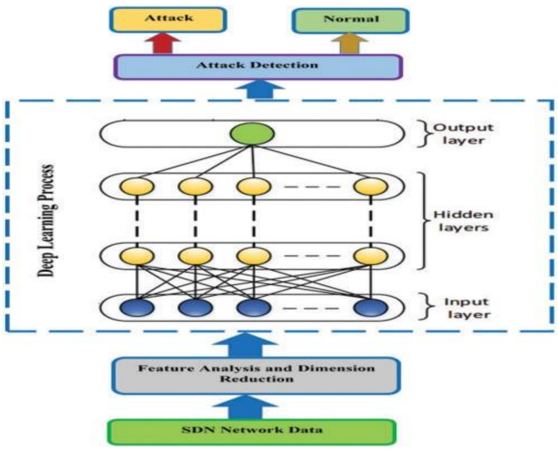


Figure 3: SDN-based deep learning model for intrusion detection

The NSL-KDD dataset is utilized in this work to assess the aftereffects of applied calculations. The dataset was made to resolve a few inborn issues with the KDD-cup 1999 dataset [48,49]. The first KDD'99 dataset was comprised of the test and train datasets that had recently been utilized to survey the presentation of IDS models. It consolidates three sorts of highlights: traffic-related, content-related, and fundamental elements. The NSL-KDD dataset is a refreshed rendition of the KDD dataset. As indicated by their attributes, assaults in the dataset are delegated U2R (User to Root) assaults, R2L (Remote to Local) assaults, testing assaults, and DoS (Denial of Service) assaults.

4 Results and Discussions

This segment represents the investigation of close outcomes, conversations and the climatic arrangement of re-enactment of grouping and discovery situations based on the proposed interruption information.

5.1 Results of the environmental simulation of the clustering algorithms based on traffic data

The game runs in WEKA weather conditions with NSL-KDD on a PC with a Core-i5 processor and 4 GB of RAM [52]. Before using the clustering system, we perform a standardization interaction on the fully inserted dataset credits in the range 0-1 and the number of clusters is set to four. The NSL-KDD dataset is split using density-based calculations, K-implies, Farthest First, Canopy, Exception-augmentation (EM), and clustering, with the training dataset containing the major types of attacks. The display of the above calculations is estimated based on the number of occasions per cluster, the running time, and the number of poorly grouped examples. The results of the tested games are displayed in tabs. 1-3 and Figs. 4-8. Groups 0 to 3, including ordinary cases, are distributed into four groupings by clustered occurrences. The bundled sample transport is divided into Normal, DoS, R2L, U2R, and Probe.

Table 1: 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.

Table 2: 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

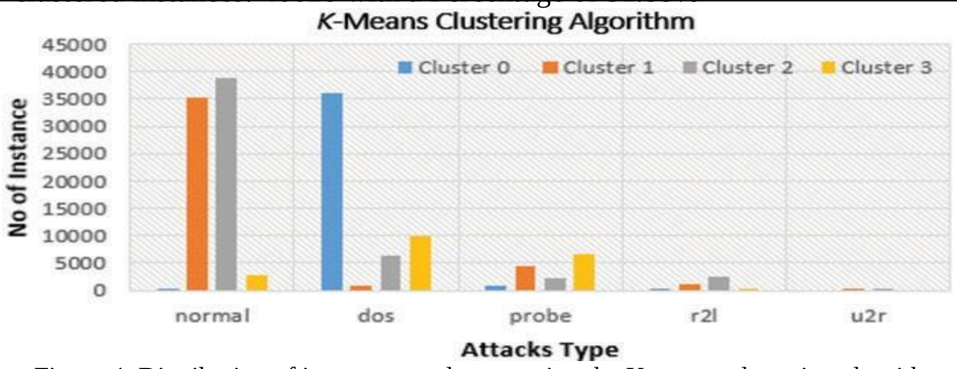
Time is taken to build a model (full training data): 0.39s

incorrectly clustered instances: 47143 with Percentage of 1.87%.

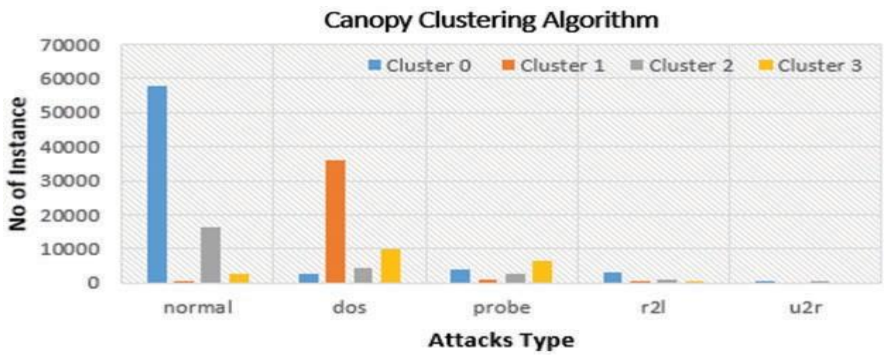
Table 3: The simulation results of the canopy clustering algorithm

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%



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



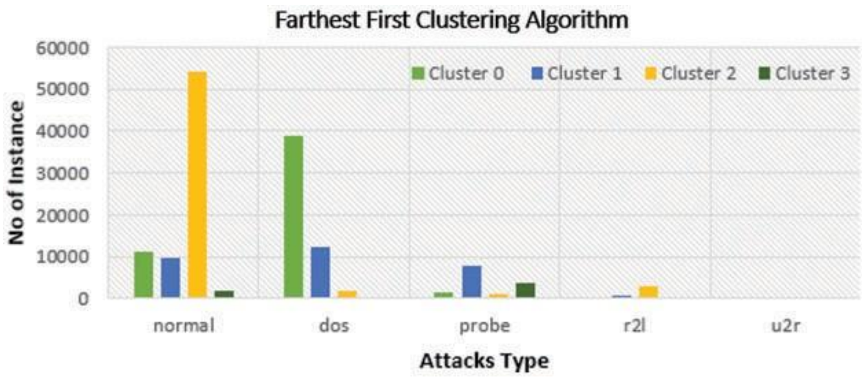
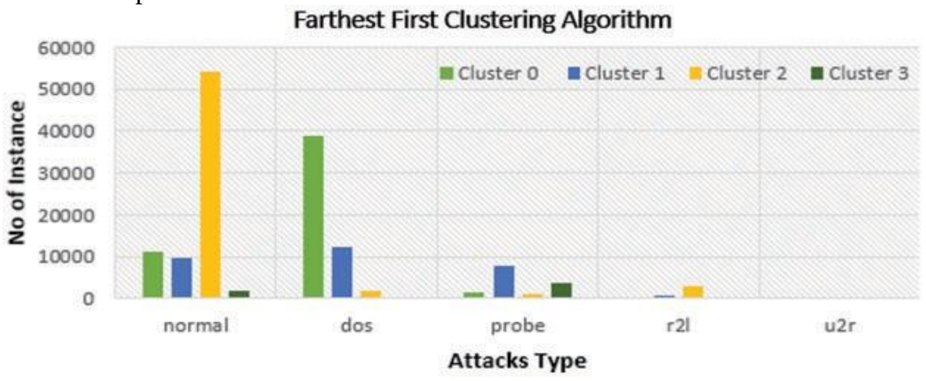


Figure 5: Distribution of instances to clusters using the farthest first clustering algorithm

The after effects of the grouped occurrences utilizing the K-implies calculation are displayed in Fig. 4. Tab. 4 shows the quantity of bunched examples for each tried group. Besides, this table shows the dissemination of each assault's examples. The K-implies conspire requires 4.51 seconds to assemble bunch models, and the quantity of inaccurately grouped occurrences is 65108. Figure 5 portrays the bunched occurrence results acquired utilizing the Farthest First calculation. Tab. 5 presents the quantity of grouped cases in each tried bunch. This table additionally shows the appropriation of each assault's occasions. The Farthest First plan requires 0.39 seconds to construct group models, and the quantity of mistakenly bunched examples is 47143.



The Canopy calculation brings about bunched cases, as displayed in Fig. 6. Tab. 6 shows the quantity of bunched cases for each tried group. Besides, this table shows the dispersion of each assault's occurrences. The Canopy plot requires 2.53 seconds to fabricate group models, and the quantity of erroneously bunched occasions is 46628. Figure 7 portrays the aftereffects of grouping examples utilizing the Exception-amplification (EM) calculation. Tab. 7 shows the quantity of grouped occasions for each tried bunch. This table additionally shows the conveyance of each assault's examples. The EM conspire requires 40.48 seconds to construct group models, and the quantity of erroneously bunched cases is 36667.

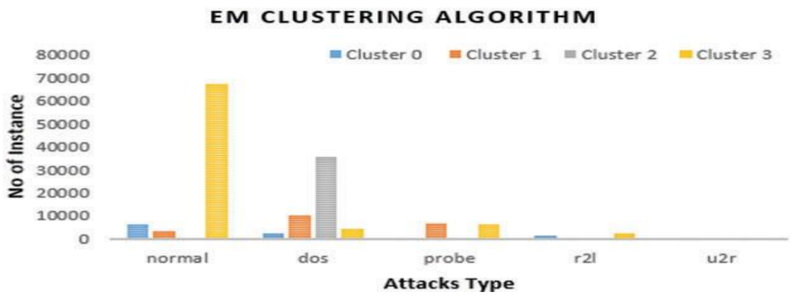


Figure 6: Distribution of instances to clusters using the canopy clustering algorithm

Figure 8 portrays the bunched case results acquired with the Density-based grouping calculation. Tab. 4 presents the quantity of bunched cases in each tried group. Moreover, this table shows the dissemination of each assault's occasions. The thickness based grouping plan requires 5.41 seconds to assemble bunch models, and the quantity of inaccurately bunched occasions is 48184. Tab. 5 and Fig. 9 analyze the five bunching calculations in view of the quantity of occurrences in every one of the four groups. The execution season of the five calculations is analyzed in Tab. 6 and Fig. 10. A correlation of five bunching calculations in view of the quantity of erroneously grouped occurrences is displayed in Tab.7 and Fig. 11.

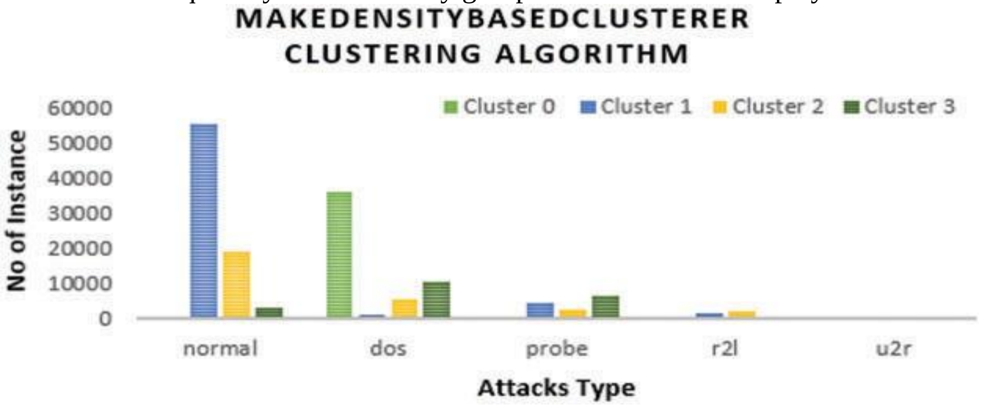


Figure 7: Distribution of instances to clusters using the EM clustering algorithm

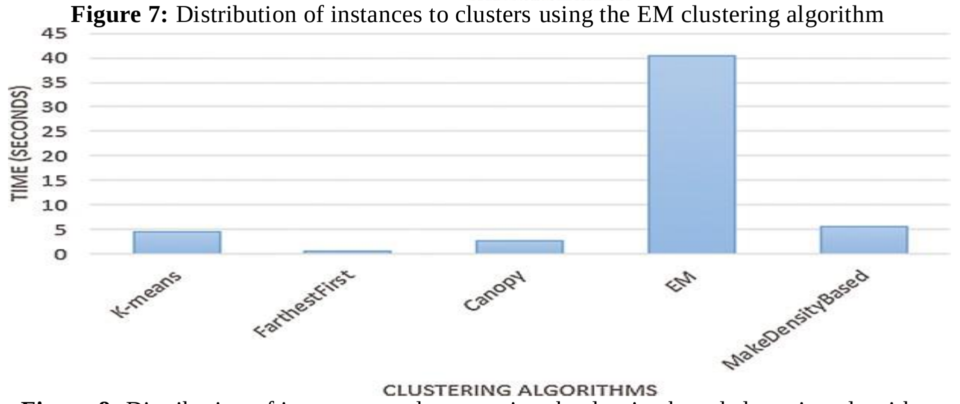


Figure 8: Distribution of instances to clusters using the density-based clustering algorithm

The entirety of the reproduction and examination results show that the dissemination of occasions differs starting with one bunch then onto the next. The Farthest First bunching calculation likewise has the most limited execution season of the five grouping calculations. Besides, among the five calculations, the EM bunching calculation creates the best number of erroneously grouped occurrences.

Table 5: 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

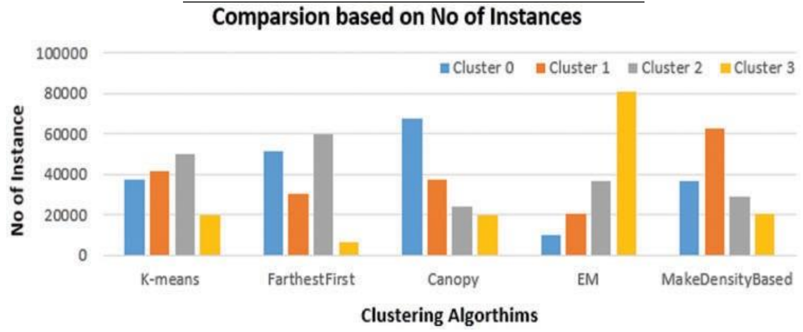


Figure 9: Comparison between clustering algorithms based many of instances

Table6: Comparison of between the clustering algorithms based on the 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%

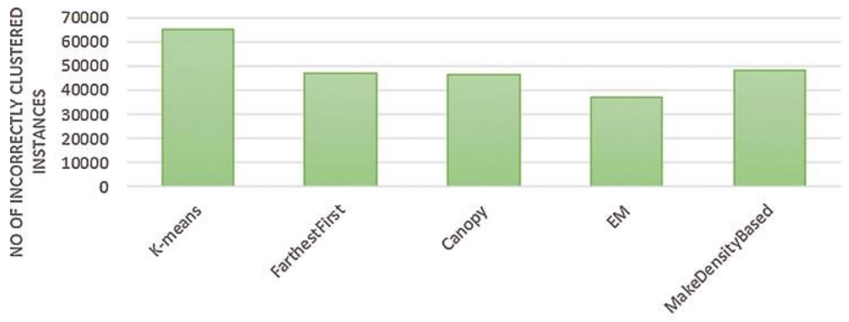


Figure 11: Comparison between the five clustering algorithms based on the number of incorrectly clustered instances

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

For binary class attack classification, the deep learning-based IDS model is used. It determines whether the input data is normal or malicious. The classification accuracy is evaluated for all 41 features, and the proposed deep learning model outperformed other machine learning techniques such as LR, LDA, and NB in terms of detection of different attack classes, as shown in Tab. 7 and Fig. 12.

Table 7: 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%

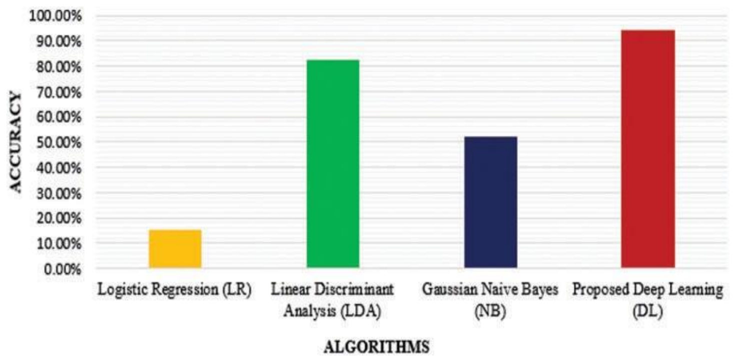


Figure 12: Comparison of the proposed DL algorithm with different algorithms

The training percentage in the proposed model is 20%, the number of epochs is 200, the batch size is 10, and the number of selected attributes is 16. Figures show the accuracy and loss percentages for various types of epochs. 13 and 14, where increasing the number of epochs increases the accuracy percentage while decreasing the loss percentage.

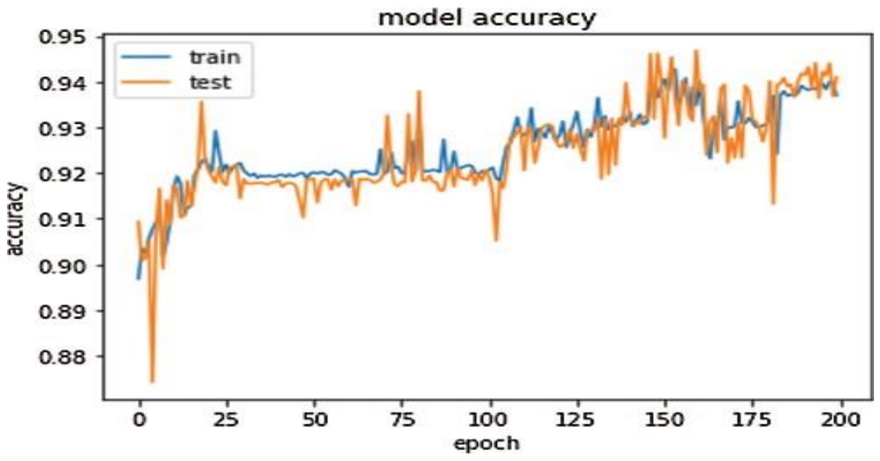


Figure 13: Proposed deep learning model accuracy.

5 Conclusion and Future Work

Perceiving and recognizing mysterious dangers and dangers is a critical task for providing a secure SND framework using a competent interrupt identification framework in recent times. Consequently, using density-based clustering calculations, farthest prime, canopy, exception enhancement (EM), and K-implicated cluster, this paper provides a cluster-based examination of the NSL dataset. -KDD. In addition, it features a deep learning framework to promote a powerful interrupt identification framework that can recognize obscure, ill-conceived and harmful exercises. The reproduction results show that the use of the Farthest First plan resulted in a preferential appropriation of occurrence over the other four clustering plans. Furthermore, while distinguishing the opportunities for interruption, the deep learning model surpasses existing estimates in terms of accuracy. The proposed DL strategy, for example, achieved high recognition accuracy of 94.21%. We intend to use advanced artificial intelligence tools in our future work to recognize new forms of attack on the SDN organization. Additionally, a certifiable run of an IoT-based SDN will be created and scheduled in online security applications.

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 CommunicationsSurveys&Tutorials,vol.18,no.1,pp.602–622,2015.
2. J.Ali,B.Roh,B.Lee,J.OhandM.Adil,“Amachinelearningframeworkforpreventionofsoftware-defined networking controller from DDoS attacks and dimensionality reduction of big data,” in Proc. IEEE Int.Conf.onInformationandCommunicationTechnologyConvergence(ICTC),Jeju,SouthKorea,pp.515–519, 2020.
3. J.AliandB.Roh,“AneffectivehierarchicalcontrolplaneforsoftwaredefinednetworksleveragingTOPSIS forend-to-endQoSclass-mapping,”IEEEAccess,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.KarakusandA.Durresi,“Asurvey:Controlplanescalabilityissuesandapproachesinsoftware-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 softwaredefinednetworking(SDN)controllers,”inProc.WorldCongressonComputerApplicationsand InformationSystems(WCCAIS),Hammamet,Tunisia,pp.1–7,2014.
7. M. Karakus and A. Durresi, “Quality of service (QoS) in software defined networking (SDN): Asurvey,”Journal of Network and Computer Applications, vol. 80, no. 4, pp. 200–218, 2017.
8. H.Kwon,“Defendingdeepneuralnetworksagainstbackdoorattackbyusingde-triggerautoencoder,”
9. IEEE Access, vol. 9, pp. 2169–3536, 2021.
10. 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.
11. 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.
12. 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.
13. 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.

14. 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.
15. E. Hemdan and D. Manjaiah, "Digital investigation of cybercrimes based on big data analytics using deep learning,"inProc. Deep Learning and Neural Network s:Concepts, Methodologies,Tools,andApplications, 1st edition, vol. 2, USA: IGI Global, pp. 615–632,2020.
16. J.JabezandB.Muthukumar,"Intrusiondetectionsystem(IDS):Anomalydetectionusingoutlierdetection approach,"ProcediaComputerScience,vol.48,no.7,pp.338–346,2015.
17. S. Duque and M. Omar, "Using data mining algorithms for developing a model for intrusion detection system(IDS),"ProcediaComputerScience,vol.61, pp.46–51,2015.
18. 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.
19. Y. Liu, Y. Kuang, Y. Xiao and G. Xu, "SDN-based data transfer security for internet of things," IEEE InternetofThingsJournal,vol.5,no.1,pp.257–268,2017.
20. W.El-ShafaiandE.Hemdan,"Robustandefficientmulti-levelsecurityframeworkforcolormedicalimages intelehealthcareservices,"JournalofAmbientIntelligenceandHumanizedComputing,vol.7,no.2,pp.1–16, 2021.
21. O. Al-Jarrah and A. Arafat, "Network intrusion detection system using attack behavior classification,"in
22. Proc. IEEE Int. Conf. on Information and Communication Systems (ICICS), Irbid, Jordan, pp. 1–6, 2014.
23. 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.
24. A. Alarifi, S. Sankar, T. Altameem, K. Jithin and W. El-Shafai, "A novel hybrid cryptosystem for secure streamingofhighefficiencyH.265 compressedvideosinIoTmultimediaapplications,"IEEEAccess,vol. 8, pp. 128548–128573,2020.
25. P. Sangkatsanee, N. Wattanapongsakorn and C. Charnsripinyo, "Practical real-time intrusion detection usingmachinelearningapproaches,"ComputerCommunications,vol.34,no.18,pp.2227–2235,2011.
26. A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusiondetection,"IEEECommunicationsSurveys&Tutorials,vol.18,no.2,pp.1153–1176,2015.
27. S.Ibrahim,M.Egila,H.Shawky,M.ElsaidandW.El-Shafai,"Cancelablefaceandfingerprintrecognition basedonthe3Djigsawtransformandopticalencryption,"MultimediaToolsandApplications,vol.79, no. 19, pp. 14053–14078,2020.
28. B. Yang, X. Fu, N. Sidiropoulos and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learningandclustering,"inProc.Int.Conf.onMachineLearning,Sydney,Australia,pp.3861–3870,2017.
29. W. Lin, S. Ke and C. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearestneighbors,"Knowledge-BasedSystems,vol.78, no. 4,pp.13–21,2015.
30. Z. Wang, "Deep learning-based intrusion detection with adversaries," IEEE Access, vol. 6, pp. 38367– 38384,2018.
31. 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.
32. U. Shafique and H. Qaiser, "A comparative study of data mining process models (KDD, CRISP-DM and SEMMA),"InternationalJournalofInnovationandScientificResearch,vol.12,no.1,pp.217–222,2014.
33. S. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: Areview,"
34. Applied Soft Computing, vol. 10, no. 1, pp. 1–35, 2010.
35. I. Raghav, S. Chhikara and N. Hasteer, "Intrusion detection and prevention in cloud environment: A systematicreview,"InternationalJournalofComputerApplications,vol.68,no.24,pp.7–11,2013.
36. 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.
37. L. Deng and D. Yu, "Deep learning: Methods and applications," Foundations and Trends in Signal Processing, vol. 7, no. 4, pp. 197–387,2014.
38. W. El-Shafai,S.El-Rabaie, M. El-Halawany and F. Abd El-Samie, "Efficient hybrid watermarking schemes forrobustandsecure3D-MVCcommunication,"InternationalJournalofCommunicationSystems,vol.31,no.4,pp.1–18,2018.
39. 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.
40. Y. Lin, H. Lei, X. Li and J. Wu, "Deep learning in NLP: Methods and applications," Journal of University ofElectronicScienceandTechnologyofChina,vol.46,no.6,pp.913–919,2017.
41. Y. Hande, A. Muddana and S. Darade, "Software-defined network-based intrusion detection system," in Proc.Innovationsin ElectronicsandCommunicationEngineering, Springer,LectureNotesinNetworksand Systems, vol. 7, Singapore: Springer, pp. 535–543,2017.