Measurement: Sensors 24 (2022) 100527

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Recurrent nonsymmetric deep auto encoder approach for network intrusion detection system

Srikanth yadav M. [a,*], R. Kalpana [b]

[a] Department of Information Technology, Vignan's Foundation for Science Technology & Research (Deemed to be University), Vadlamudi, Guntur, AP, India
[b] Department of Computer Science and Engineering, Puducherry Technological University (Erstwhile Pondicherry Engineering College), Puducherry, India

ARTICLE INFO

Index Terms:
Deep learning
Anomaly detection
Autoencoders
NSLKDD
CICIDS
Network security

ABSTRACT

An important part of network security is a network intrusion detection system (NIDS). In the face of the need for new networks, there are issues regarding the feasibility of traditional approaches. More directly, these difficulties are connected to the increasing degrees of human contact required and the diminishing levels of detection precision. A new deep learning intrusion detection approach is presented in this research to overcome these problems. The recurrent non-symmetric deep autoencoder we've suggested for learning unsupervised features is described here (RNDAE). A new deep learning classification model based on LightGBM RNDAEs is also shown. NSL-KDD, CICIDS2017, and CSECICIDS2018 datasets were used to evaluate our proposed classifier in TensorFlow. If our model holds up, it has the potential to be used in the latest generation of network intrusion detection systems (NIDS).

1. Introduction

Network intrusion detection systems have several obstacles, the most important of which is ensuring their long-term stability and dependability. Despite substantial developments in NIDS technology, the bulk of NIDS systems [1] still relies on less capable signature-based detection approaches rather than anomaly detection methods. High false-error rate, difficulty in acquiring valid training data [2], data longevity, and system dynamics all contribute to this inability to move [3]. Because of the current state of affairs, relying on these techniques would only lead to poor detection shortly. With this challenge, we want to develop a technique for anomaly detection that can outpace the rapid changes in modern networks while still being universally accepted. Three key limits are mostly to blame for this network security problem. In the first place, the amount of network data is expected to continue to expand at a rapid pace.

In recent years, as more people have access, IoT devices have become more popular, and cloud-based services have become more generally recognized, this industry has risen significantly. Data analysis methods must become quicker, more reliable, and more exact to deal with big numbers. Another consideration is how much detail and monitoring are necessary to increase efficiency and accuracy. For NIDS data analysis to move away from high-level abstractions, a more detailed and contextual approach is required. Behavioral changes can be traced back to network components such as operating system versions or protocols, for example. The last problem is that today's networks are overflowing with so many different protocols and data types. Because it makes it difficult to distinguish between normal and abnormal behavior, this may be the most severe hindrance. As a result, it is more difficult to maintain an exact standard, and the risk of manipulation or zero-day attacks is increased [4].

Machine learning techniques including Naive Bayes, Decision Trees, and Support Vector Machines have recently been applied in NIDS research. In general, the use of these methods has increased the precision with which anomalies have been detected [5]. Data analysis, such as recognizing meaningful data and patterns, requires a high level of human experience because of these technologies' limitations. Not only does this require a lot of time and money, but it is also susceptible to human error. In a varied and sophisticated situation, a large amount of training data may be challenging.

Deep learning is being studied as a possible solution to the limitations stated above [6]. There are ways around shallow learning that use this more sophisticated subset of machine learning. Deep learning may be able to overcome shallow learning approaches due to its better layer-wise characteristics, according to early studies. It allows for a more thorough investigation of network data and the quicker detection of

* Corresponding author.
E-mail address: srikanthyadav.m@gmail.com (S. M.).

https://doi.org/10.1016/j.measen.2022.100527
Received 4 August 2022; Received in revised form 19 September 2022; Accepted 13 October 2022
Available online 14 October 2022
2665-9174/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

abnormalities. New deep learning models for NIDS activity in modern networks are presented in this research. It is possible to analyze a wide variety of network traffic using a model that combines deep and shallow learning.

On top of Random Forest, we've proposed the Recurrent Non-symmetric Deep Auto-Encoder, or RNDAE, which we believe is superior to RF [7,8]. TensorFlow was used to run our model on the NSL-KDD, CICIDS2017, and CSECICIDS2018 datasets, and we got promising results. The dataset's faults are known, yet they are nonetheless often used as a benchmark for comparable efforts, allowing us to make precise comparisons.

The proposed model offers the following contributions.

- Non-symmetric data dimensionality reduction for unsupervised feature learning is provided by the RNDAE approach instead of typical autoencoder techniques. Thus, our method is capable of producing better classification results than those typically employed.
- RF classification is used with RNDAEs to create a unique classifier model. Deep learning and shallow learning approaches can be used in conjunction to decrease analytical overheads. This study's outcomes are on par with or better than those of other research in the field while using less training time.

Listed below are the sections of this document. Contextual information may be found in Section 2. Section 3 focuses on the current state of knowledge. In Section 5, we take a closer look at our solution, which is detailed in Section 4 of the report. In Section 6, we explain the findings of our evaluation. Conclusions are included in Section 7 of the study.

2. Background

In this section, we will present the context needed to comprehend our objectives and the concepts underlying the approach proposed in this work.

2.1. Challenges

Anomaly detection, forensics, and security have all made substantial use of network monitoring. Many new challenges have arisen for NIDSs due to recent technological developments. Considerable concerns include, but are not limited to.

- *Accuracy:* The precise levels described above are not attainable with the technologies currently in use. As a result, more granularity, depth, and understanding of context are critical to providing a complete and accurate picture. Aside from the obvious inconvenience, there are also significant financial [10,12], computational, and practical [13] costs associated with this.
- *Attacks with low frequency:* Attacks such as these have repeatedly eluded prior anomaly detection approaches, such as those involving artificial intelligence. Attacks of this nature, which occur infrequently, are harder to identify by NIDS due to the imbalances in the training dataset [11].
- *Compliance:* Modern networks have included several new technologies to lessen their dependency on aging infrastructure and the management practices that go along with it. Virtualization and software-defined networking (SDN) [15] are becoming increasingly popular in today's network infrastructures. NIDS must be able to adapt to these new technologies and their impacts.

2.1.1. Deep learning

Deep learning is a subfield of machine learning that takes machine learning closer to artificial intelligence [16]. Using several levels of representation makes it simpler to convey complex concepts and relationships Unsupervised [17] and supervised [14] learning algorithms are used to create higher levels of abstraction, based on the output characteristics from lower levels.

We recommend using an autoencoder [18,19] because it is widely utilized in deep learning research. Using an unsupervised neural network, an autoencoder learns the optimal parameters for recreating its output as closely as possible to its input. In comparison to Principal Component Analysis (PCA) [20], it is capable of providing a more robust and non-linear generalization.

As a result of using backpropagation [21], the final values are equal to their initial ones. That is to say, it makes an effort to learn as close as possible to the true identity function. An auto-encoder typically has an input layer, an output layer, and a hidden layer. In most cases, the input dimension is smaller than the layer that is being hidden. As may be seen in Fig. 1, an autoencoder is depicted.

Using autoencoders as a nonlinear [22] transformation of their networks to reveal new data structures and comparing their findings with PCA is standard procedure among academics. These methods are built on the encoder-decoder [23] paradigm. As a result, the amount of data that has to be entered is reduced. An initial translation into a lower-dimensional space is followed by an expansion into the original data. Finally, when a certain number of layers have been trained, each layer's code is fed into the next one. The deep auto-encoder structure contains a unique coding layer at its heart for this purpose. Compression of feature vectors for classification or stacking in an autoencoder may be achieved by using this coding layer.

The hidden layer can be used to shrink large datasets. By reducing the dimensionality of the data distribution, the auto-encoder is driven to focus on the most critical characteristics. Ideally, the features generated by the auto-encoder will better reflect the data points than the raw data.

3. Existing work

There is a lot of interest in the topic of deep learning, and it is currently being used in a wide range of disciplines, including healthcare [24], automotive design [25], manufacturing [26], and even the police force.

In addition, several previously published publications are concerned with NIDS. It is in this part that the most current and interesting findings will be highlighted.

Several comparable publications in the literature investigate machine learning for intrusion detection [27]. Using a deep learning technique, intrusion detection systems may be built using deep learning ideas. Machine learning techniques for intrusion detection systems are being examined, according to a framework for software analysis. Intrusion detection systems are being evaluated for their ability to identify deep learning algorithms, according to an analysis of available methods. There is evidence that machine learning approaches are being

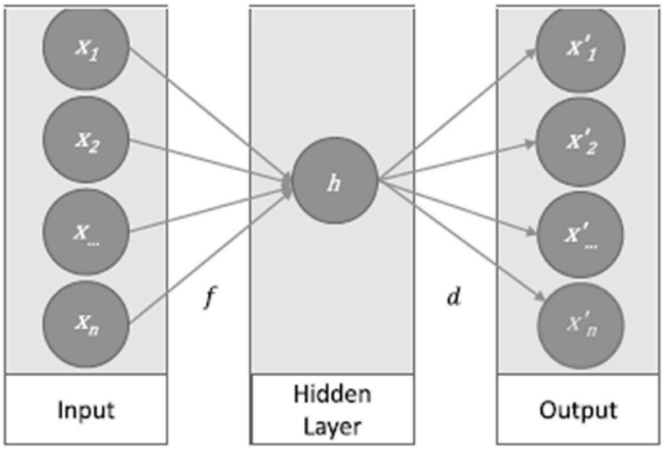

Fig. 1. A simple autoencoder.

tested for intrusion detection equipment. The intrusion detection system data sets utilized by IDSs recommend research based on these data sets.

Research on intrusion protection databases has recently been published by Ref. [28] authors. 34 datasets and 15 characteristics have been outlined in detail. There are five types of environmental monitoring apps: general information, evaluation, data quantity and quality, and environmental monitoring.

Intrusion detection applications are mentioned in a paper on deep learning methods [29]. In addition, any mining and learning tools deployed by the intrusion detection algorithm were made more complex by the study. The data used in this investigation was divided into three types: packets, Net Flow data, and publicly available data sets [30]. conducted an online analysis of intrusion detection techniques in comparative studies (IoT). The classification was made based on the recognition systems, the positioning of IDS, and the safety risks.

Analysis of existing systems, including workloads, metrics, and procedures relevant to each standard evaluation parameter, provided by Ref. [31] authors [32]; authors developed the RBC-IDS methodology for the research of IDS viability. In the RBC-IDS comparison, adaptive learning techniques had a prediction rate of 99.12% and an accuracy rate of 99.91%, respectively. RBC-IDS was proposed as a method by the authors.

Also, a real-time and computer-efficient anomaly detection tool has been presented by Ref. [17] to find causal connections between subsystems utilizing feature extraction algorithms and time series partitions. Free energy assaults are detected using an anomaly index based on the DBN principle and Boltzmann computer-dependent learning methods. TPR accuracy of 98% and less than 2% FPR are achieved by the proposed device.

A deep learning-based strategy for constructing a scalable and effective NIDS has been proposed by Ref. [33]. An auto-encoder and Softmax regression were used to create the NIDS. An intruder data collection of the benchmark network was tested using NSL-KDD.

4. Proposed methodology

4.1. Data pre-processing

OneHotEncoder and min-max normalisation are the primary components of the pre-processing scheme given here. In the pre-processing, you may encounter data that isn't present in the dataset if you use the unlabeled raw as an input. Whenever we confront difficulties, we must be prepared to handle them. Missing values, strategy, and axis can all be sent to the Imputer class. For categorization variables with no such ordinal connection, the integer coding is insufficient. For insufficient execution or unexpected outcomes, it is possible for models to envision a usual order across categories by using this encoding. An onehotencoder can be used to encode the integer's representation in this case. Here the encoded integer variable is removed, and a new binary variable is placed for each unique integer value. The datasets are normalized using the min-max normalisation to create unique data pieces.

4.1.1. Feature selection using LightGBM

LightGBM is a decision tree-based gradient boosting framework that improves model efficiency while consuming less memory. Gradient-based One-Side Sampling and Exclusive Feature Bundling are two unique strategies used to overcome the constraints of the histogram-based algorithm employed in all Gradient Boosting Decision Tree frameworks. The LightGBM Algorithm has the following characteristics: GOSS and EFB. To put it another way, they form the basis of the model and provide it an advantage over alternative GBDT frameworks.

The information gain may be computed using a variety of different data examples. As gradient sizes increase, the amount of added information increases as well. Maintaining accurate information gain estimation is critical to GOSS, thus it discards those occurrences with minor gradients at random. When the range of the information gain is vast, this method, rather than uniformly random sampling, can provide a more precise estimate of gain. The procedural code is shown in Fig. 2.

4.1.1.1. Classification. Here, we offer new RNDAE autoencoders that don't use symmetrical hidden layers and instead use non-symmetric layers. An encoder phase is proposed as a replacement for encoding and decoding paradigms. Thus, it is possible to reduce computational and temporal overheads while ensuring excellent accuracy and efficiency.

To extract hierarchical unsupervised features from huge datasets, NDAE can be used. It learns non-trivial features using a training method similar to that of a standard autoencoder. Fig. 3 depicts the structural differences between an autoencoder and an RNDAE.

Our experiment's methods should be shown using the proposed design. We sent in a request for training and testing data sets. A friendly or focused data collection is used for training, while a neutral or attack data collection is utilized for assault. Afterward, we'll standardize the information we've accumulated thus far. A deep learning algorithm was used to teach the data once it had been standardized and then educated. Our IDS model was able to identify the assault since we used standardization to collect data during the testing procedure. Fig. 4 depicts the proposed architecture in all its glory.

Before processing begins, the unlabeled raw data is presented, and you'll find that there is no data in the resulting dataset. We need to be ready for everything that may come our way in the future. The Imputer class may pull in more arguments besides only the ones for determining missing values, approach, and axis. If there is no ordinal relationship between the variables, then the Integer's coding is inappropriate. This encoding has the potential to lead to sparse execution or surprising outcomes for the model by allowing it to envision a conventional order between categories.

Using min-max standardization, the datasets for this investigation will be transformed into specific data types. Detecting intrusions is one of the most challenging tasks for an intrusion detection system. We intend to reduce the complexity of the KDD cup 99 training datasets by removing standard data. The proposed pre-processing work improves false-positive rates and increases the overall power of the system. Categorical data are encoded, and numeric properties are normalized, as part of the recommended approach's pre-processing module.

5. Datasets

1) NSL-KDD dataset: In NSL-KDD, all of KDDCUP99's data is combined into a single archive. Using the KDDCUP99 benchmark data, the NSL-KDD data collecting team examines the issues raised. Each NSL-KDD link record has 41 attributes designated as either standard or an attack, with one specific sort of attack. When it comes to NSL-KDD training, there are 22 different attack kinds to choose from, and there are an additional 17 attack types available just for testing purposes. Table 1 shows the distribution of the NSL-KDD dataset.

DoS Attack: An assault known as denial-of-service (DoS) slows down or completely shuts down an organization's network, making it impossible to carry out legitimate business operations. Neptune, Smurf, Pod, and Teardrop are all that stand for this.

Probe Attack: Data on the network can be accessed by another assault using the Probe attack. This sort of attack gathers information about the target system before launching an attack to circumvent security measures. Portsweep, IPsweep, Nmap, and Satan are just a few examples.

Remote-to-Local (R2L) Attack: As a local user, an attacker can transmit a packet across the network and exploit vulnerabilities like Password Guess, FTP-Write, Imap, and Phf. to get access to a remote workstation that they don't have permission to access.

User-to-Root (U2R) Attack: In this case, an attacker can get root access to a server by impersonating a normal user and taking advantage of vulnerabilities in the system. A few instances of Perl-based exploits are

```
# Importing Required Library
import pandas as pd
import lightgbm as lgb
# Similarly LGBMRegressor can also be imported for a regression model
from lightgbm import LGBMClassifier
# Reading the train and test dataset
data = pd.read_csv("cancer_prediction.csv)
# Removing Columns not Required
data = data.drop(columns = ['Unnamed: 32'], axis = 1)
data = data.drop(columns = ['id'], axis = 1)
# Skipping Data Exploration
# Dummification of attack Column (1-Benign, 0-attack)
data['attack']= pd.get_dummies(data['attack'])
# Spliiting Dataset in two parts
train = data[0:i]
test = data[i-1:n]
# Separating the independent and target variable on both data set
x_train = train.drop(columns =['attack'], axis = 1)
y_train = train_data['attack']
x_test = test_data.drop(columns =['attack'], axis = 1)
y_test = test_data['attack']
# Creating an object for model and fitting it on training data set
model = LGBMClassifier(model = LGBMClassifier()
model.fit(x_train, y_train)
# Predicting the Target variable
pred = model.fit(x_test)
print(pred)
accuracy = model.score(x_test, y_test)
print(accuracy)
```

Fig. 2. LightGBM pseudocode for feature selection.

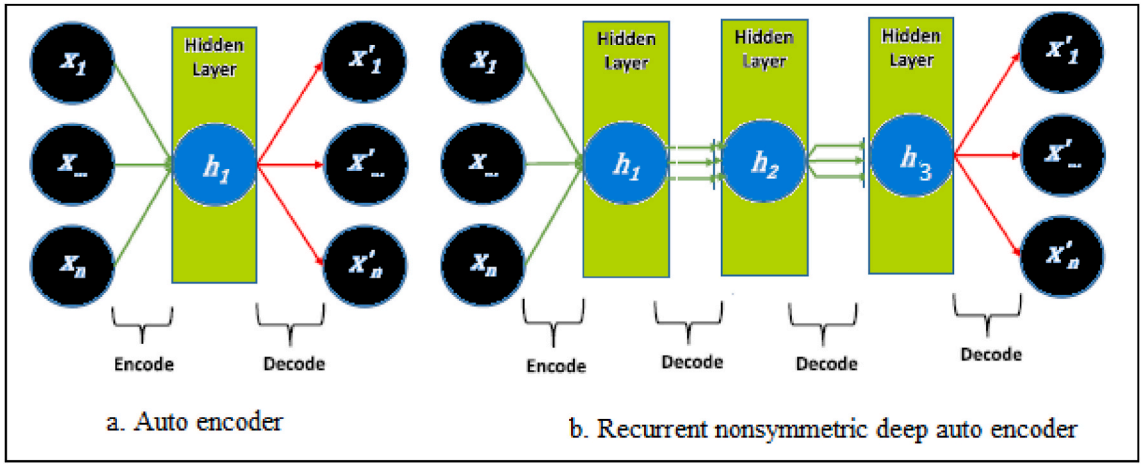

Fig. 3. Autoencoder vs RNDAE.

Bug-overflow, Load-module, and Spy.

2) CICIDS2017 Dataset: Real-world data is used to create the CICIDS2017 dataset, which explains why the dataset contains only benign and current attacks (PCAPs). Source and destination IPs, source and destination ports, protocols, and attacks are all included in the findings of the CICFlowMeter. Additionally, the definitions of the extracted characteristics are supplied. When creating this dataset, we paid careful attention to including realistic background traffic. As a result of our B-Profile technology's ability to offer genuine, innocuous background traffic, we've discovered how people engage with each other. For this dataset, a total of 25 fictitious users were created using HTTP, HTTPS, FTP, SSH, and email protocols.

Benign: With "RandomUnderSampler," an Imblearn balancing library [9] python function, 270,000 traffic records (BENIGN) are sampled to address the imbalance problem. A minimum of 5000 records for each attack type is now required for this classifier to be effective in detecting assaults with low numbers of records of difficulties with multi-class

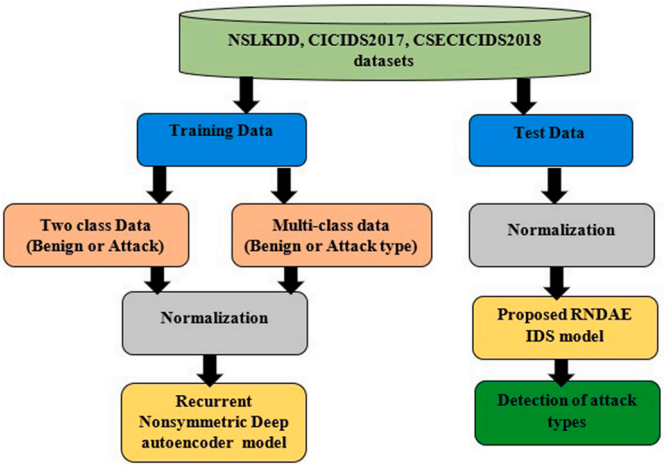

Fig. 4. Proposed IDS architecture.

Table 1
Distribution of the NSL-KDD dataset.

Attack type	Flow count	Training	Test
Normal	12717530	67412	9736
DoS_Attack	659149	45915	7471
Probe_Attack	281201	11683	2409
R2L_Attack	114589	987	2743
U2R_Attack	201901	51	202
Total	13974370	126048	22561

identification (MCID). Python's SMOTE module is utilized to accomplish this task.

Brute force attacks: Because they appear to get into accounts with bad username/password combinations, brute force assaults on networks are persistent. The third option was to use a central server brute force assault dictionary to generate an SSH and MySQL account.

DoS Attacks: It has been demonstrated that a single attacker machine may shut down web servers when using the strong tools Slowloris and LOIC. It begins with Slowloris, a device that creates a full TCP connection to a distant server. When the tool is running, it periodically sends incomplete HTTP requests to the server to maintain open sockets. There will be no more new connections possible because of the restricted connection handling capabilities of each web server. It's also possible to perform DoS assaults on a website by using the HOIC software that's widely available.

DDoS Attacks: High Orbit Ion Cannon (HOIC), sometimes referred to as HOIC, is an open-source network stress test and denial of service attack tool designed to simultaneously impact up to 256 URLs. The Low Orbit Ion Cannon designed by Praetox Technologies was meant to be replaced. To launch a distributed denial-of-service assault, we're utilizing the free HOIC utility on four different machines.

Botnet: A backdoor will be put on the victim's PC if they are successful in breaking into their system. Searching and destroying any other infected devices on the internal network is what we'll be doing with his PC. The CICIDS2017 dataset's distribution can be seen here. Keystroke recording and form capture is usually used to collect financial information by the "man in the browser," however they can be used for harmful and criminal purposes. Ransomware Crypto-Locker, a ransomware variant, takes advantage of it as well for its distribution. The open-source Ares botnet, which is also utilized as an add-on, is capable of the following.

Infiltration: This method involves sending a malicious file to the target through email and then taking advantage of a security flaw in the target's software to compromise the system. If the breach is successful, a backdoor will be placed on the victim's computer. Infected devices on the internal network will be found and attacked using his PC. The distribution of the CICIDS2017 dataset is shown below Table 2.

3) CSECICIDS2018: DDoS data was collected by the University of New Brunswick and used in this dataset. You can access the entire dataset here. New versions of this dataset will be accessible at the website provided above if and when they are made available. Numerous DDoS attacks were recorded against the university's servers while the dataset was available to the public. To determine whether or not the sent packets are malicious, it is crucial to keep the Label column in mind while developing machine learning notebooks for this data.

Data is divided into files based on dates. The notebook designer has to partition the dataset into balanced files to generate better predictions. Table 3 illustrates the CSECICIDS2018 dataset's distribution.

6. Results and discussions

Deep learning researchers utilize TensorFlow to develop their suggested classification model. On a Windows 10 64-bit PC with 16 GB of RAM, TensorFlow GPU-enabled, we ran all of our experiments. NSL-KDD, CSECICIDS2017, and CSECICIDS2018 were used in our assessments. The NIDS community considers these two datasets to be gold standards. It is much easier to compare results from various research and approaches when there are publicly available datasets to do so. Table 4 compares the experimental results with well-established approaches such as DBN and DNN and highlights the differences.

The proposed model has shown better accuracy over existing deep learning models. The performance of LightGBM + RNDAE approach on NSL_KDD has observed 96.5% accuracy. Notably, the model offered 97.8% accuracy on CICIDS2017 dataset, and 98.8% accuracy on CSE-CICIDS2018 dataset. The model accuracy is increased from 96.5% to 97.8%, and 98.8% on NSL_KDD, CICIDS2017, and CSECIDIDS2018 datasets respectively.

7. Conclusion & future work

As part of our research, we've examined some of the difficulties that existing NIDS approaches face. In response to this, our RNDAE technique for unsupervised feature learning was developed. We've created a new classification model using RNDAEs and the RF algorithm.

Using TensorFlow to construct our model has been carefully tested, and we are satisfied with the results. NSL-KDD, CICIDS2017, and CSE-CICIDS2018 datasets were used in our assessments, and our findings were quite good. Our model was evaluated on both benchmark datasets, and its classification accuracy was shown to be consistent across both sets. When it comes to accuracy, precision, and memory recall, we have

Table 2
Distribution of the CICIDS2017dataset.

Attack type	Flow count	Training	Test
SSH_Attack	239	185	40
FTP_Attack	612	487	102
XSS_Attack	186589	7535	1866
Web_Attack	193370	1552	3750
SQL_Injection_Attack	79	68	11
Hulk_Attack	465653	18558	4678
SlowHTTPTest_Attack	139982	56147	14006
Slow_Loris_Attack	10934	4365	1091
Goldeneye_Attack	41499	16522	42062
HOIC_Attack	686114	27435	6870
LOIC_UDP_Attack	1738	1358	341
LOIC_HTTP_Attack	576289	23139	5752
BoT_Attack	286187	11391	2971
Infiltration	161940	6469	1629
Benign	12697529	50995	12678
Total	15448754	226206	97847

Table 3
Distribution of the CSECICIDS2018 dataset.

Attack type	Flow count	Training	Test
Benign	13484708	134850	134849
HOIC1	686012	129558	129558
LOIC-HTTP1	576191	99430	99431
Hulk1	461912	72612	72613
Bot	286191	72599	72600
FTP-BruteForce	193360	72268	72267
SSH-BruteForce	187589	47024	47024
Infiltration	161934	20703	20703
SlowHTTPTest1	139890	4954	4954
GoldenEye1	41508	2999	865
Slowloris1	10990	2999	285
LOIC-UDP1 1	730	2999	114
Brute Force-Web	611	2999	43
Brute Force-XSS	230	2999	27
SQL Injection	87	2999	27
Total	16231943	671992	655360

Table 4
Performance comparison of IDS datasets.

Dataset	Model	Accuracy	Specificity	Sensitivity	F-Score
NSL_KDD	DBN	0.961	0.966	0.956	0.946
	DNN	0.951	0.961	0.937	0.924
	LightGBM + RNDAE	0.965	0.968	0.959	0.951
CICIDS2017	DBN	0.974	0.979	0.969	0.959
	DNN	0.964	0.974	0.95	0.937
	LightGBM + RNDAE	0.978	0.981	0.972	0.964
CSECICIDS2018	DBN	0.984	0.989	0.979	0.969
	DNN	0.974	0.984	0.96	0.947
	LightGBM + RNDAE	0.988	0.991	0.982	0.974

found that our way is the most effective. Our RNDAE model was compared to the usual DBN technique. Our model may increase accuracy by up to 5% while simultaneously lowering training time by up to 98.81%, according to these experiments.

Our model's capability to cope with zero-day attacks will be assessed and expanded in future work. Following our previous assessments, we'll use real-world backbone network traffic to prove that our expanded model works.

CRediT authorship contribution statement

Srikanth yadav M.: Conceptualization, Methodology, Data curation, Writing – original draft, Validation. **R. Kalpana:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Transactions on Emerging Topics in Computational Intelligence 2 (1) (2018) 41–50, https://doi.org/10.1109/TETCI.2017.2772792.
[2] H.L.Y. Bengio, P. Lamblin, D. Popovici, Greedy layer-wise training of deep networks, Proceedings of the Advances in Neural Information Processing Systems (2007), https://doi.org/10.7551/mitpress/7503.003.0024.
[3] D.P. Kingma, M. Welling, Auto-encoding variational bayes, Accessed: Apr. 11, 2021. [Online]. Available: https://arxiv.org/abs/1312.6114v10, Dec. 2014.
[4] S. Shaukat, et al., Intrusion Detection and Attack Classification Leveraging Machine Learning Technique, in: Proceedings of the 2020 14th International Conference on Innovations in Information Technology, IIT 2020, Nov. 2020, pp. 198–202, https://doi.org/10.1109/IIT50501.2020.9299093.
[5] R.R. Reddy, Effective Discriminant Function for Intrusion Detection Using SVM, 2016, pp. 1148–1153.
[6] L. Deng, O. M. Way, D. Yu, and O. M. Way, "Deep Learning : Methods and Applications".
[7] J. Coronel Gaviro, A. Boukhamla, CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed, Int. J. Inf. Comput. Secur. 1 (1) (2021) 1, https://doi.org/10.1504/IJICS.2021.10039325.
[8] N. Farnaaz, M.A. Jabbar, Random forest modeling for network intrusion detection system, Procedia - Procedia Computer Science 89 (2016) 213–217, https://doi.org/10.1016/j.procs.2016.06.047.
[9] NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/nsl.html. (Accessed 11 April 2021).
[10] IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/ids-2017.html. (Accessed 5 August 2021).
[11] IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/ids-2018.html. (Accessed 30 May 2022).
[12] G. Zhao, C. Zhang, L. Zheng, Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network, 2017, https://doi.org/10.1109/CSE-EUC.2017.119.
[13] HACKMAGEDDON – Information Security Timelines and Statistics. https://www.hackmageddon.com/. (Accessed 30 May 2022).
[14] B.A. Pratomo, Unsupervised Approach for Detecting Low Rate Attacks on Network Traffic with Autoencoder, in: International Conference on Cyber Security and Protection of Digital Services, Cyber Security), 2018, pp. 1–8.
[15] D. Kreutz, F. M. v Ramos, and P. Verissimo, "Towards Secure and Dependable Software-Defined Networks," pp. 1–6.
[16] Y. Deng, Y. Jiao, B.L. Lu, Driver Sleepiness Detection Using LSTM Neural Network, in: Security and Communication Networks, 2017, https://doi.org/10.1155/2017/4184196.
[17] H. Karimipour, A. Dehghantanha, R.M. Parizi, K.K.R. Choo, H. Leung, A Deep and Scalable Unsupervised Machine Learning System for Cyber-Attack Detection in Large-Scale Smart Grids, IEEE Access (2019), https://doi.org/10.1109/ACCESS.2019.2920326.
[18] M.E. Aminanto, R. Choi, H.C. Tanuwidjaja, P.D. Yoo, K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection IEEE Transactions on Information Forensics and Security, 13 3 (2017) 621–636, https://doi.org/10.1109/TIFS.2017.2762828. Oct.
[19] Y. Yu, J. Long, Z. Cai, Network Intrusion Detection through Stacking Dilated Convolutional Autoencoders, Security and Communication Networks (2017), https://doi.org/10.1155/2017/4184196.
[20] F. Salo, A.B. Nassif, A. Essex, Dimensionality reduction with IG-PCA an ensemble classifier for network intrusion detection, Comput. Network. 148 (Jan. 2019) 164–175, https://doi.org/10.1016/J.COMNET.2018.11.010.
[21] J. Malik, A. Akhunzada, I. Bibi, M. Imran, A. Musaddiq, S.W. Kim, Hybrid deep learning: an efficient reconnaissance and surveillance detection mechanism in SDN, IEEE Access 8 (2020) 134695–134706, https://doi.org/10.1109/ACCESS.2020.3009849.
[22] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, Science (1979), https://doi.org/10.1126/science.1091277, 2004.
[23] F. Farahnakian, J. Heikkonen, A Deep Auto-Encoder Based Approach for an Intrusion Detection System, in: International Conference on Advanced Communication Technology, ICACT, 2018, pp. 178–183, https://doi.org/10.23919/ICACT.2018.8323688, 2018-Febru.
[24] Z. Qiu and Y. Zhu, "A novel structure of blockchain applied in vaccine quality control: double-chain structured blockchain system for vaccine anticounterfeiting and traceability," J. Healthc. Eng., vol. 2021, 2021, DOI: 10.1155/2021/6660102.
[25] A. Taylor, S. Leblanc, N. Japkowicz, Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks, 2016, https://doi.org/10.1109/DSAA.2016.20.
[26] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine Learning and Deep Learning Approaches for CyberSecurity: A Review," IEEE Access, vol. vol. 10. Institute of Electrical and Electronics Engineers Inc., pp. 19572–19585, 2022. DOI: 10.1109/ACCESS.2022.3151248.
[27] R. Singh, H. Kumar, R.K. Singla, An intrusion detection system using network traffic profiling and online sequential extreme learning machine, Expert Syst. Appl. (2015), https://doi.org/10.1016/j.eswa.2015.07.015.
[28] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, A. Hotho, A survey of network-based intrusion detection data sets, Comput. Secur. (2019), https://doi.org/10.1016/j.cose.2019.06.005.
[29] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutorials, (2016), https://doi.org/10.1109/COMST.2015.2494502.
[30] B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in Internet of Things, J. Netw. Comput. Appl. (2017), https://doi.org/10.1016/j.jnca.2017.02.009.

[31] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, B.D. Payne, Evaluating computer intrusion detection systems: a survey of common practices, ACM Comput. Surv. (2015), https://doi.org/10.1145/2808691.

[32] B. Deore, S. Bhosale, A decisive approach to intrusion detection system using machine learning model, WEENTECH Proceedings in Energy (Mar. 2021) 143–154, https://doi.org/10.32438/WPE.152021.

[33] A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system, Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (2016), https://doi.org/10.4108/eai.3-12-2015.2262516.