IMPROVED DETECTION NETWORK BASED IDS BY CONOVLUTION NEURAL NETWORK WITH SVM -RBF KERNEL

Ayush Gautam[1] ,Er. Shivani Rana[2]
Faculty Mtech Department, Himachal Pradesh Technical University , Hamirpur

ABSTRACT

Attackers are always drawn to valuable and important information, which is why the network is always under assault. For example, a malicious packet might be sent to a user's machine, where it can be modified or altered, before being sent across the network for illicit purposes known as an attack. Intrusion An attack on the system server arises because of a flaw or vulnerability in the system, such as user error or incorrect setup. The combination of numerous vulnerabilities may potentially be used to make an intelligent incursion.They can be subclassified based on their ability to identify a signature, specification or abnormality. Attacks are detected if a device or network connections analyses an attack against a signature in the inner database of an intrusion detection system (IDS). This article uses CNN and SVMs to enhance classification accuracy by 10%, precision by 2%, and sensitivity by 3%. These findings show that the proposed technique considerably improves due to the efficient mapping of characteristics.

Keywords: IDS, CNN, Network, Deep learning

DOI Number: 10.14704/nq.2022.20.9.NQ44198 **Neuro Quantology 2022; 20(9):1718-1724**

INTRODUCTION

With the rapid expansion of applications, roughly 20.4 billion devices will be online in 2020, increasing to 75 billion per month of 2025. Different sensors installed in devices enable remote data acquisition and processing in real time. The data collected from sensors enables them to create intelligent decision-making platforms and successfully manage environments [2,3,16,17,18]. Users have access and operate their gadgets from everywhere, at any moment, exposing them to a variety of risks. (1) Unauthorized access to and exploitation of personal information; (2) encouraging attacks on some other systems; and (3) rising security vulnerabilities [6,7,19]. IDSs are necessary to keep networks secure and capable of detecting intrusions. Due to the restricted computational and power resources available to devices (bandwidth, battery, memory, and computing), a sophisticated Intrusion Detection System (IDS) can indeed be created. It is critical to advance research in this area of computer network intrusion detection [1,8,15,23,24]. Denial of service (DoS) is a severe and destructive attack that prevents genuine consumers from obtaining the data for that they have paid, in violation of the terms of the Service Level Agreement (SLA), resulting in massive financial losses for businesses and organisations. Additionally, denial-of-service attacks affect tiny networks, like smart homes, intelligent health systems, and intelligent agricultural systems [12]. DoS attacks that disrupt important, intelligent services like healthcare can also result in human death due to the disruption of routine services. devices (for example, intelligent refrigerators, smart televisions, and air conditioners) are easily attacked by attackers who exploit their weaknesses to launch denial-of-service attacks [10,20,31]. As a result, one of the primary concerns for researchers all over the world is the protection of these devices. Globally, intrusion detection is being studied to fix this matter. IDS are classified into three categories based on their detection capabilities: Depending on a signature, a specification, or an anomaly. Whenever a device or network connections analyses an attack against a signature contained in the inner IDS database, an attack is identified by IDSs. If a device or network operation matches one of the saved signatures/patterns, a warning will be generated. This method is extremely reliable and effective at recognizing identified risks, and its process is simple to comprehend. However, this technique is ineffective in classifying new attacks and discrepancies between current These sorts of assaults do not have a meaningful signature to identify them [4,5,17]. If the divergence from a specified behaviour profile exceeds a predefined threshold, an anomaly-based intrusion detection system (IDS) issues an alert. Classifying intrusions does not seem to follow a typical pattern, and understanding the whole spectrum of normal activity is not an easy task. Emerging threats may be identified using this method. As a consequence, there is a significant rate of false positives with this method. Routing tables, protocols, and nodes, for example, are all part of the specification-based approach since they are all defined by a set of rules and criteria. It is possible to identify intrusions when network behaviour deviates from standards' specifications, using specification-based techniques. Therefore, specification-based detection is used for the same goal as anomaly detection: to separate aberrant behaviour from normal behaviour. Nevertheless, the specification-based method differs significantly from the other two approaches in that each specification's rules must be manually created by a human expert. Anomaly-based identification tends to have higher false-positive rates than criteria provided manually. There is no need for training for detection systems based on specifications, as the process begins the moment a specification is written [12].

As global commerce grows, so does the need for secure information in both the workplace and everyday life. The Internet is increasingly being used in both these contexts. A computer network is now at the heart of every global structure. When it comes to protecting sensitive data, network security has become a must in today's world. In order to protect sensitive data from unauthorised access, it is essential to identify network intrusions [2].

In order to anticipate network assaults, a well-organized intrusion detection system is a must-have for network protection. An Intrusion Detection model based on feature selection allows us to reduce the size of the NSL-KDD data set while still incorporating machine learning to create an Intrusion Detection model that can be used to detect attacks on structures and improve intrusion detection by utilising captured data[2]. The model's goal is to create an intrusion detection system that can identify new and previous undiscovered assaults using prime signatures and characteristics of known attacks as the number of new attacks grows. [5].Nahida Islam et al. [1] recognised numerous forms of risks and discussed both shallow (such as RF, SVM, and DT) and deep deep learning (DBN, DNN, LSTM, bidirectional LSTM (Bi-LSTM), and stacked LSTM based IDS in the context. Five benchmark datasets, including NSL-KDD, IoTDevNet, IoT, DS2OS, and IoTID20 as well as the Botnet database, were used to test the efficiency of these models.

RELATED WORK

Khalid Albulayhi et al. [2] conduct a review of DL methodologies for IDSs in the IoT and accompanying datasets with the goal of finding gaps, vulnerabilities, and a neutral reference design. A comparative examination of IDSs is presented, along with a discussion of anomaly-based IDSs on deep learning approaches, including unsupervised, supervised, and hybrid approach. Each of these 3 categories of approaches has mostly been used to contexts.

Cao Tien Thanh [3]. The research provides an overview of the technology, focusing on various types of attacks and anomalies, and also their detection via an adaptive IDS. This paper presents an in-depth examination and evaluation of many deep learning and machine learning-based network IDS. Additionally, the study identifies a series of research challenges in order to facilitate further advancements in methods for resolving specific difficulties.

Zeeshan Ahmad, et.al [4]. This article first defines IDS and then presents a taxonomy based on prominent machine learning and deep learning approaches used in the creation of network-based IDS (NIDS) systems. A complete assessment of recent NIDS-related articles is presented, together with a discussion of the proposed solutions' strengths and weaknesses. The article next discusses recent trends and improvements in machine learning and deep learning-based NIDS in terms of suggested method, dataset selection, and evaluation metrics. The researchers highlighted numerous research obstacles and suggested future directions for study on enhancing ML and DL-based NIDS by highlighting the inadequacies of the presented approaches.

Ghada Abdelmoumin, et.al [5]. The purpose of this article is to discuss the application of optimization approaches to improve the performance of single-learner AML-IDS models, like 1-SVM AML-IDS and PCA models, for the purpose of developing scalable, efficient, and distributed intelligent IDS for identifying intrusions. The researchers assess these AML-IDS models using Microsoft Azure Machine Learning Studio (AMLS) and two datasets comprising benign and malicious and industrial (IIoT) traffic on the network. Geeta

Singh and Neelu Khare [6] This article summarises publicly available labelled intrusion datasets and machine learning approaches. Following that, a brief description of the literary works in which machine learning techniques are used to implement NIDS in various networking contexts is provided, including cloud networks, conventional networks, WSNs, and Ad-Hoc networks. As a result, this study combines publicly available intrusion databases and machine learning approaches used in contemporary IDS to shed light on current issues and future possibilities.

Javed Ashraf, et.al [7]. This study seeks to provide an in-depth examination of the technologies, protocols, architecture, and hazards associated with compromised devices, as well as an overview of intrusion detection methods. Additionally, this paper analyses several deep learning and machine learning-based strategies for detecting systems compromised by cyber-attacks.

PROPOSED WORK

collect the dataset from" https://www.stratosphereips.org/datasets-iot23" and preprocess the dataset by normalizing the features. After preprocessing input in convolution network and mapping on nonlinear space by RELU and sigmoid function. This step runs till optimize the features are.For optimization, use gradient descent and optimize the loss function features optimize, then learning by SVM classifier and make classifier model.

Test model and analysis precision, recall, and accuracy Neural networks (NNs) are a type of computer network. LeCun and his colleagues (1988) LeCun et al. [63] are a subset of fully connected multi-layer perceptrons that use weight sharing to process input with a grid-like architecture (for example, iimages). The spatial correlation of a signal is used by CNNs to limit the design in a more logical way. Their design, which is based mostly on biological visual system, has two essential characteristics that make them ideal for image processing: spatial pooling and spatially shared weights. Shift-invariant features, i.e., filters that seem to be useful over the entire image frame, are learned by these types of networks (owing to a stationary nature of picture statistics). The pooling layers are in charge of lowering the output's sensitivity to minor input shifts and distortions.

A value function is presented to evaluate how beneficial it is for an agent to use policy π to visit state's'. The value is indeed the mathematical expectation of return, and the Bellman expectation formula is used to approximate the value:

$$V^{\pi}(s_t) = \mathbb{E}[r_{t+1} + \gamma V^{\pi}(s_{t+1})] \ldots\ldots\ldots\ldots\ldots (1)$$

The expectation term of $V^{\pi}(s_t)$, also known as the state-value function, can be extended as a product of transition probability, policy, and return as described in the following:

$$V^{\pi}(s_t) = \sum_{\alpha_t \in \mathcal{A}} \pi(\alpha_t|s_t) \sum_{s_{t+1} \in S} \mathrm{T}\,(s_{t+1}|s_t, \alpha_t)[R(s_t, s_{t+1}) + \gamma V^{\pi}(s_{t+1})] \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The Bellman equation is the name of this equation. Bellman equation can be written as follows where the agent always chooses the action based on the optimal policy π^* that maximises the value:

$$V^{\pi}(s_t) = \max_{\alpha_t} \sum_{s_{t+1} \in S} \mathrm{T}\,(s_{t+1}|s_t, \alpha_t)[R(s_t, s_{t+1}) + \gamma V^{\pi}(s_{t+1})] \stackrel{\Delta}{=} \max_{\alpha_t} Q^*(s_t, \alpha_t) \ldots\ldots\ldots\ldots (3)$$

Because the complexities of the actual world, getting an optimal value function V* does not have enough knowledge to rebuild an optimal strategy π^*. As a result, a quality function (Q-function) is set out as follows:

$$Q^{\pi}(s_t, \alpha_t) = \sum_{s_{t+1}} \mathrm{T}\,(s_{t+1}|s_t, \alpha_t)[R(s_t, s_{t+1}) + \gamma V^{\pi}(s_{t+1})] \ldots \ldots \ldots \ldots (4)$$

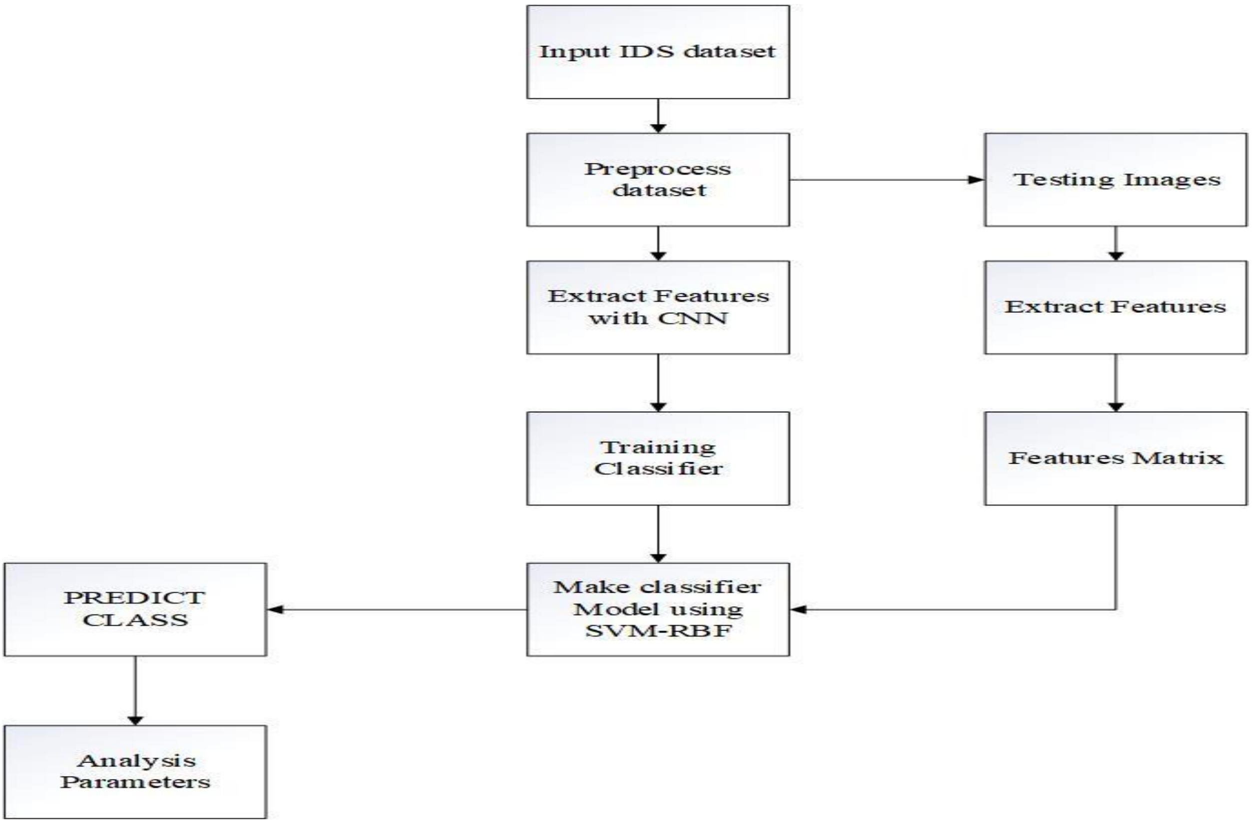

Fig 3.4 Proposed Flow Chart

Result Analysis

In experiment use dataset with 41 features , In proposed and existing approach use SVM but in proposed approach use CNN based feature mapping before SVM apply. In experiment different performance parameters like accuracy, precision recall and F-score use for models evaluation. Apart of this Analysis the true positive, false positive values of model.

```
[ ] print("For Random Forest:")
    TP_old, FP_old, TN_old, FN_old = perf_measure(Y_test, BNB_Classifier.predict(X_test))
    print ("TP:", TP_old, "\tFP:", FP_old, "\t\tTN:", TN_old, "\tFN:", FN_old)

    print()
    print("ForCNN -> Random Forest:")
    TP_new, FP_new, TN_new, FN_new = perf_measure(Y_test_sub, DTC_Classifier.predict(X_test_sub))
    print ("TP:", TP_new, "\tFP:", FP_new, "\t\tTN:", TN_new, "\tFN:", FN_new)

    print()
    print("For CNN+RF:")
    tp = TP_old +TP_new
    fp = FP_new
    tn = TN_old +TN_new
    fn = FN_new
    print ("TP:", tp, "\tFP:", fp, "\t\tTN:", tn, "\tFN:", fn)

    For Random Forest:
    TP: 2981        FP: 188                 TN: 3872        FN: 517

    ForCNN -> Random Forest:
    TP: 508         FP: 13          TN: 175         FN: 9

    For CNN+RF:
    TP: 3489        FP: 13          TN: 4047        FN: 9
```

Fig4.1 : Comparison of Confusion matrix parameters in code snippet

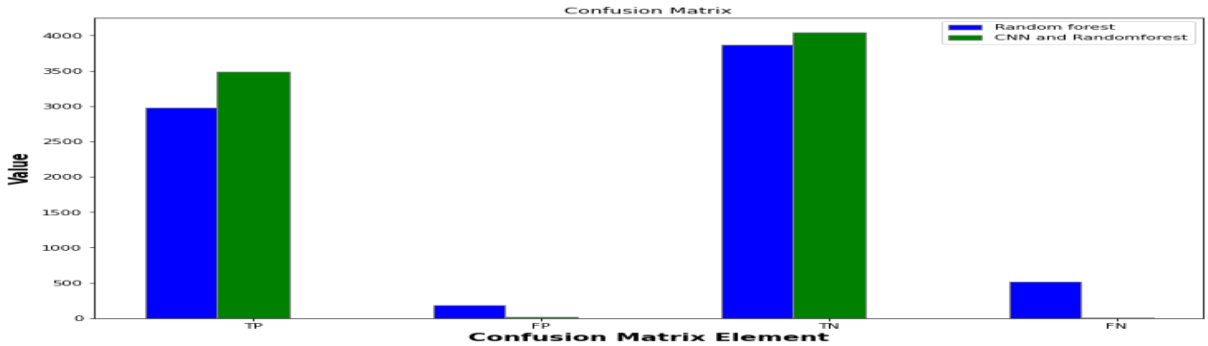

Fig4.2: Comparison of Confusion matrix parameters of Proposed and Existing

In figure4.1 and fig 4.2 performance analysis of confusion matrix .In graphs shows TP and TN improve significantly in proposed approach and reduce significantly FP and FN , so its effect on Fig4 where analysis the performance by accuracy ,precision, recall and f-score. So its also improve in CNN based method compare to existing approach

```
acc_old= (TP_old + TN_old) / (TP_old + FP_old + TN_old + FN_old)
mis_old= (FP_old + FN_old) / (TP_old + FP_old + TN_old + FN_old)
prec_old= TP_old / (TP_old + FP_old)
sen_old= TP_old / (TP_old + FN_old)
spec_old= TN_old / (TN_old + FP_old)

acc= (tp + tn) / (tp + fp + tn + fn)
mis= (fp + fn) / (tp + fp + tn + fn)
prec= tp / (tp + fp)
sen= tp / (tp + fn)
spec= tn / (tn + fp)

print ("Accuracy")
print ("Old: ", acc_old, "\tNew: ", acc)
print ("\nMisclassification")
print ("Old: ", mis_old, "\tNew: ", mis)
print ("\nPrecision")
print ("Old: ", prec_old, "\tNew: ", prec)
print ("\nSensitivity")
print ("Old: ", sen_old, "\tNew: ", sen)
print ("\nSpecificity")
print ("Old: ", spec_old, "\tNew: ", spec)

Accuracy
Old:  0.906721354855782       New:  0.9970891770309606

Misclassification
Old:  0.09327864514421805      New:  0.0029108229690394285

Precision
Old:  0.9406752918901862      New:  0.9962878355225585

Sensitivity
Old:  0.8522012578616353      New:  0.9974271012006861

Specificity
Old:  0.9536945812807882      New:  0.9967980295566502
```

Fig4.3: Comparison of Different Performance in code snippet

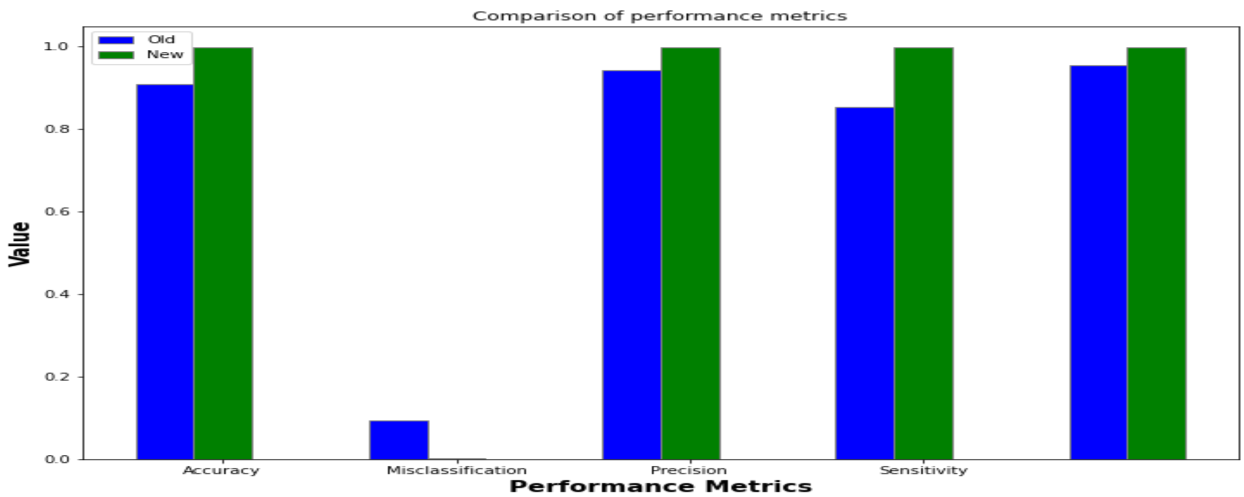

Fig4.4: Comparison of Different Performance parameters of Proposed and Existing

In fig4.3 and 4.4 Analysis the improvement in performance of proposed approach compare to existing.

```
[ ] accPercent= ((acc- acc_old)/acc_old) *100
    misPercent= ((mis_old- mis)/mis_old) *100
    precPercent= ((prec- prec_old)/prec_old) *100
    senPercent= ((sen- sen_old)/sen_old) *100
    specPercent= ((spec- spec_old)/spec_old) *100

    print ("Accuracy increase percentage: " ,accPercent, "%")
    print ("Missclasification decrease percentage: " ,misPercent, "%")
    print ("Precison increase percentage: ", precPercent, "%")
    print ("Sensitivity increase percentage: ", senPercent, "%")
    print ("Specificity increase percentage: ", specPercent, "%")

    Accuracy increase percentage:  9.966438056325693 %
    Missclasification decrease percentage:  96.87943262411348 %
    Precison increase percentage:  5.911980904763097 %
    Sensitivity increase percentage:  17.041261321704123 %
    Specificity increase percentage:  4.519628099173545 %
```

Fig4.5: Performance comparison on proposed and existing

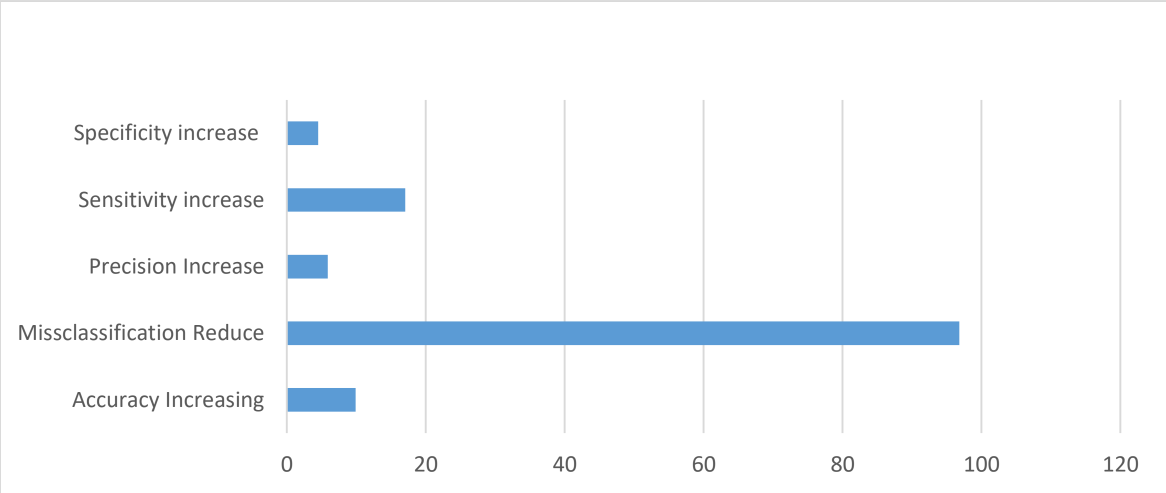

Fig4.6: Performance comparison on proposed and existing

CONCLUSION

Numerous phases necessary in creating and implementing IDS, like data pre-processing, feature reduction, model planning and implementation, and especially ML-based methodologies for IDS, enhance computational burden. Another difficulty and area for future research is building an efficient IDS with low computational requirements. To reduce future threats, it is thought that more threat detection research is required, and that security vulnerabilities, like confidentiality and privacy, have been identified and therefore must be fixed and avoided. Traditional intrusion detection methods are difficult to implement because of the IoT's diverse and scattered nature. By using the Cloud computing service, we were able to handle data from various systems in an effective manner. For mission-critical applications, responsiveness and intrusion detection are a prerequisite. They can be subclassified based on their ability to identify a signature, specification or abnormality. Attacks are detected if a device or network connections analyses an attack against a signature in the inner database of an intrusion detection system (IDS). This article uses CNN and SVMs to enhance classification accuracy by 10%, precision by 2%, and sensitivity by 3%. These findings show that the proposed technique considerably improves due to the efficient mapping of characteristics.

REFERENCE

1. Islam, N., Farhin, F., Sultana, I., Kaiser, M. S., Rahman, M. S., Mahmud, M., ... & Cho, G. H. (2021). Towards machine learning based intrusion detection in networks. *Comput. Mater. Contin*, *69*, 1801-1821.
2. Albulayhi, K., Smadi, A. A., Sheldon, F. T., & Abercrombie, R. K. (2021). Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors*, *21*(19), 6432.
3. Thanh, C. T. (2021, November). A Survey of Machine Learning Techniques for Security.

In *International Conference on Future Data and Security Engineering* (pp. 139-157). Springer, Singapore.
4. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.
5. Abdelmoumin, G., Rawat, D. B., & Rahman, A. (2021). On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things. *IEEE Internet of Things Journal*.
6. Singh, G., & Khare, N. (2021). A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. *International Journal of Computers and Applications*, 1-11.
7. Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, *9*(7), 1177.
8. Chaabouni, N. (2020). *Intrusion detection and prevention for systems using Machine Learning* (Doctoral dissertation, Université de Bordeaux).
9. Rani, D., & Kaushal, N. C. (2020, July). Supervised machine learning based network intrusion detection system for Internet of Things. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.
10. Kiran, K. S., Devisetty, R. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building a intrusion detection system for environment using machine learning techniques. *Procedia Computer Science*, *171*, 2372-2379.
11. Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Computer Science*, *167*, 1561-1573.
12. Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686-1721.
13. Bhati, B. S., & Rai, C. S. (2020). Analysis of support vector machine-based intrusion detection techniques. *Arabian Journal for Science and Engineering*, *45*(4), 2371-2383.
14. Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An adaptive ensemble machine learning model for intrusion detection. *IEEE Access*, *7*, 82512-82521.
15. Rahman, M. A., Asyhari, A. T., Leong, L. S., Satrya, G. B., Tao, M. H., & Zolkipli, M. F. (2020). Scalable machine learning-based intrusion detection system for IoT-enabled smart cities. *Sustainable Cities and Society*, *61*, 102324.
16. Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for applications. *Wireless Personal Communications*, *111*(4), 2287-2310.
17. Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. *Journal of Network and Computer Applications*, *161*, 102630.
18. da Costa, K. A., Papa, J. P., Lisboa, C. O., Munoz, R., & de Albuquerque, V. H. C. (2019). Internet of Things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, *151*, 147-157.
19. Tabassum, K., Ibrahim, A., & El Rahman, S. A. (2019, April). Security issues and challenges in IoT. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-5). IEEE.