International Journal of Scientific Research & Engineering Trends
Volume 8, Issue 4, July-Aug-2022, ISSN (Online): 2395-566X

Performance Analysis of Intrusion on Detection Using Machine Learning Techniques

Ishita Bansal
Computer science SRN International School Jaipur Rajasthan.
Email.bansalishita1011@gmail.com

Abstract- As cyber attacks become more common, cyber security is quickly becoming one of the most important things for every company to worry about. Artificial Intelligence (AI) and Machine Learning (ML), especially Deep Learning (DL), can be used as key enablers for cyber-defence because they can help find threats and even tell cyber analysts what to do next. This makes it possible to use these technologies as key tools for cyber-defence. For AI and ML to be used more quickly in cyber security and for effective cyber defence systems to be made, the private sector, academic institutions, and the government need to work together on a global scale. In this research, we look into the different deep learning techniques that are used to find network intrusions, and we present a DL framework that can be used in a variety of cyber security applications. Machine learning is being used more and more in many different fields because it has been shown to work better than traditional algorithms that are based on rules. Different cyber-detection systems are currently adding these techniques in order to help the first level of security analysts or even replace them in the long run. Even though the goal of fully automating detection and analysis is appealing, machine learning needs to be looked into very carefully to see if it can help with cyber security. We go over some of the ways that machine learning has been used to find intrusion, malware, and spam. This analysis is for people who work in security. The goal is twofold: first, to figure out how mature these solutions are right now, and second, to find out what the main problems are with these solutions that keep them from being used right away in machine learning cyber detection strategies. Our conclusions are based on a thorough review of the relevant research that has already been published, as well as the results of experiments done on real enterprise systems and real network traffic.

Index Terms: Machine Learning, Artificial Intelligence, Deep Neural Networks, Cybersecurity, Intrusion Detection Systems.

I. INTRODUCTION

According to Cisco [1], by 2023, there will be three times as many IP-connected devices as people on the planet, resulting in up to 4.8 ZB of IP traffic annually. This rapid growth creates enormous security challenges. Consequently, learning to recognise network attacks is a crucial issue that must never be disregarded. Intrusion Detection is now the primary technology for network security since it is a significant and proactive security mechanism. The goal of intrusion detection systems (IDSs), which can be used on a misuse or anomaly basis, is to spot anomalous access or attacks on internal network security [2]. The popularity and use of machine learning (ML) are expanding. Existing approaches are being refined, and their capacity to comprehend and address actual problems is highly valued. These successes have prompted machine learning to be used in a variety of industries, including computer vision, medical analysis, gaming, and social media marketing. Machine learning methods are sometimes preferable to conventional rule-based algorithms or even human operators. This trend is also having an impact on the realm of cyber security, as some detection systems are getting ML upgrades. First-level operators in Network and Security Operation Centres (NOC and SOC) may benefit from detection and analysis technologies based on machine learning, even though developing a fully automated cyber defence system is still a distant goal. This article, which is primarily aimed at security operators, tries to evaluate the current state of these systems' maturity, to pinpoint their current state, and to indicate certain areas in need of development.

With the development of network-based technologies, network-based systems' ability to operate reliably has become increasingly important. As computers become more and more integrated into the systems we rely on, it is crucial to have the ability to identify intrusions in computer systems. Internet security is a crucial component of an enterprise's success that has an impact on everything from business to cost management. Internet attacks that are catastrophic can stop businesses in their tracks, making security knowledge more valuable. According to recent studies, the number of network attacks has greatly increased, and as a result, experts' interest in studying network incursions has grown. It has been crucial to assess machine learning (ML) techniques for network-based Intrusion Detection Systems (IDS) throughout the previous decade and into the present.[3].

Intrusion Detection System (IDS)

An intrusion detection system (IDS) is a tool or software programme that keeps an eye out for malicious activity or rule violations on a network or in a system. Any intrusion activity or violation is often recorded centrally using a security information and event management (SIEM) system, alerted to an administrator, or both. Using alarm filtering techniques, a SIEM system aggregates the outputs from several sources to identify suspicious behaviour. IDS types can be categorised depending on small networks or extensive ones. Network intrusion detection systems (NIDS) and host-based intrusion detection systems are the two most commonly used categories (HIDS). An example of a HIDS is a system that keeps track of crucial operating system files, whereas an example of an NIDS is a system that examines incoming network traffic.

Since contemporary malware can automatically create novel variations with the same destructive consequences but posing as entirely distinct executable files, malware analysis is a highly relevant issue. Traditional rule-based malware identification techniques are defeated by this malware's polymorphic and metamorphic characteristics. Malware variant analysis and assigning them to the appropriate malware family can be done using machine learning (ML) techniques.

Detecting spam and phishing involves a broad range of tactics meant to cut down on the time loss and potential risk brought on by unwanted communications. The preferred method used today by attackers to gain entry into an enterprise network is through unsolicited emails, specifically phishing. Malware or links to hacked websites are often included in phishing emails. Because of the sophisticated evasion techniques attackers employ to get past conventional filters, spam and phishing detection is becoming more and more challenging. ML methods help enhance the spam detection procedure.

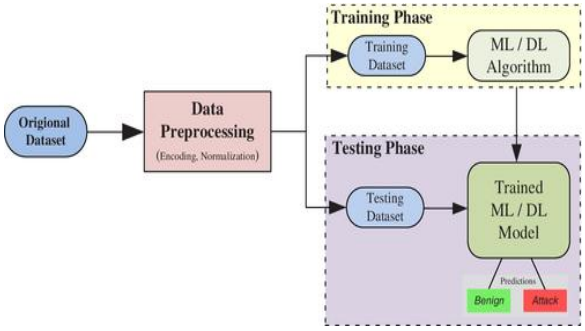


Fig 1 AI based NIDS system

Working of AI based NIDS

The following three primary steps are often involved in an NIDS based on ML and DL algorithms: Phases I of data preparation, II of training, and III of testing. Data is first preprocessed to convert it into a format that the algorithms can use. The preprocessed data is then split into the training dataset and the testing dataset at random. In the subsequent training phase, the ML or DL algorithms are trained using the training dataset. The model is tested using the testing dataset after it has been trained, and its performance is assessed based on the predictions it makes. Network traffic will be classified as either normal or attacked in the case of NIDS models.

II. RELATED WORK

An IDS based on RNNs was proposed by Yin et al. Their study's main objective was to show that RNN can be used to create classification models for binary and multiclass problems that perform better and with higher accuracy than conventional machine learning techniques. This study's experimental phase made use of the NSL-KDD dataset to assess classification accuracy. The study of the RNN-IDS model's performance for binary classification (Normal, Anomaly) and five-category classification were the two components of the experiment (Normal, DoS, R2L, U2R, and Probe). They compared the performance of the RNN model to that of an ANN, nave Bayesian, random forest, support vector machine, random tree, and multilayer perceptron in the binary classification and the five-category classification.

Li et al. : used a visual conversion of the NSL-KDD dataset to assess how well CNN performed in spotting new attacks. The data was divided into five classes throughout the experimental phase: Normal, Dos, Probe, U2R, and R2L. The fundamental features, traffic features, and content features were three categories into which the NSL-KDD feature attributes were divided. The NSL-KDD dataset had 41 features in each sample (integer, float, symbolic, or binary). They mapped different kinds of features into binary vector space and then translated the binary vector into the image to convert the NSL-KDD data format into a visual image type.

Shone et al: For unsupervised feature learning, a non-symmetric deep auto-encoder (NDAE) has been proposed. On the KDD Cup'99 and NSL-KDD datasets, stacked NDAEs were used to build the classifier model. Fundamentally, this entails the suggested switch from the symmetric encoder-decoder paradigm to relying solely on the encoder phase (non-symmetric). This is justified by the fact that, given the right learning structure, it is possible to minimise the impact on precision and efficiency while reducing computational and time overheads. A hierarchical unsupervised feature extractor called NDAE was employed. NDAEs were stacked to construct a DL hierarchy in order to actualize the model. A layer-wise unsupervised representation learning approach was made available by stacking the NDAEs, and this allowed the model to learn the intricate connections between various features.

III. MOTIVATION

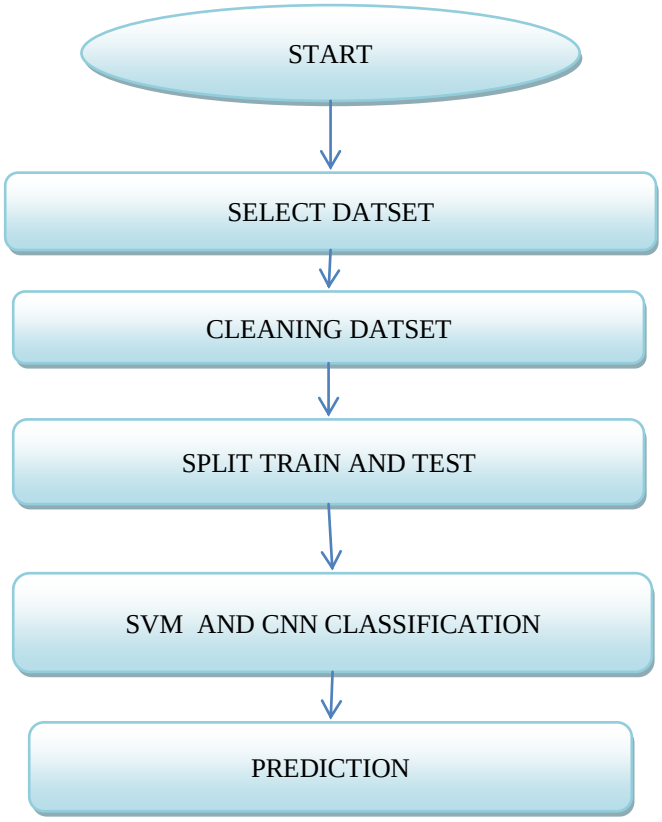
In order to increase the IDS's accuracy and comprehensiveness, a significant percentage of research in its development is devoted to developing novel system designs and detectors. Event correlation is a crucial field of study that has a strong track record in the detection of misuse. In connection with system designs, intrusion detection in wireless/mobile ad hoc and sensor networks is a growing study field. Applying ML to intrusion detection is becoming more popular since it provides flexible detectors and is well suited for anomaly detection. Hybrid systems, which include abuse and anomaly detectors, host-based and network-based modules, event correlation and stateless detectors, are now frequently created.

A significant portion of recent research on hybrid IDS has focused on the correlation of efficient alerts between the various modules[5–9]. Recent research has focused on alert aggregation, which is related to alert correlation and aims to combine related occurrences into a single generic event. The amount of warnings a system administrator has to look through and the false positive rates can both be significantly decreased as a result. Scalability is frequently addressed in research by dispersing or decentralizing the IDS. Event correlation and other detection methods based on mobile agents and synthetic immune systems can be used to accomplish this. Using Bayesian reasoning, it is decided how many probes to send out and how many tests to run in order to obtain information about the distributed system's performance. The goal of creating this system is to achieve real-time system detection while lowering computing expenses. A flaw in intrusion detection is that the IDS could end up creating a security hole.

According to many experts, state-based IDSs are more vulnerable to assaults than stateless ones because they may be swamped with events that prevent them from operating effectively. Applications of machine learning (ML) algorithms, which improve with use, are particularly susceptible to hostile attacks. In order to prevent the detection of a new planned attack, there is a risk that an adversary will influence the training process by gradually modifying the behavior over time. Several studies outline the dangers to learning algorithms itself and go through how to prevent and recognize an assault. [10]

IV. PROPOSED SYSTEM

The process of monitoring and analysing events that occur in a computer or networked computer system is known as intrusion detection. Detection is accomplished by analysing user behaviour that contradicts the system's intended use. Even if the computer is not connected to the Internet, any user will be vulnerable to intrusion. If the computer is left unattended, any intruder can gain access and attempt to exploit the system. If the computer is connected to a network, particularly the Internet, the problem becomes much worse. Remote access to the computer is possible from anywhere in the world. An intruder may attempt to gain access to important private or confidential information or launch an attack to bring the system to a halt or render it inoperable. Architecture: The main focus here is on what could be called a detection module, which would exist within a larger IDS framework. The architecture is critical, especially in wireless and mobile ad hoc networks[11]. This includes deciding where to deploy the intrusion detection system, which is considered a research challenge.



Data collection: Many researchers have used the KDD Cup ’99 data set for their work since data collection is not required. It is necessary to collect data from the environment in which IDS is to be deployed. This also includes a process of labelling data for supervised learning.

Data preprocessing: Data preprocessing is necessary mainly for the detection methodologies that are suitable for ML. Although the availability of the data set is very convenient for researchers, the challenge is in the transformation applied.

Performance: There are several methods that can be adopted to help achieve better performance in IDS. This could be in terms of detection rates, usage of memory usage, feature selection, and sampling of data. In relation to data transformation, various transformations and feature sets may aid in improved intrusion detection.Different

ways of preprocessing the data are available, which may yield improved performance.

Other issues: Detecting new intrusions is always a challenge. This is due to new software being available, which inevitably has vulnerabilities that can be exploited. Therefore, re-training IDS is necessary once a new data set is available. When and how this is to be done, and whether to have online training or unsupervised learning, is still an open research challenge.

Feature Selection: Another challenge in developing IDS is to obtain feature set that is comprehensive enough to separate normal data from intrusive data by keeping the size of the data set as small as possible. More the features more difficult it is to detect intrusions. For many ML algorithms increasing the number of features may significantly increase the training time required to learn the intrusion task. Also the run time will slow down and memory requirements will increase with more features, commonly referred to as 'the curse of dimensionality' [12-15] Hence, much research has been devoted to developing efficient techniques to perform feature selection

The proposed model is introduced to overcome all the disadvantages that arise in the existing system. This system will increase the accuracy of the classification results by classifying the data based on the social network mental disorders and others using random forest classification algorithm. It enhances the performance of the overall classification results.

Data Selection and Loading

- Data selection is the process of selecting the data for detecting attacks.
- In this project, the KDDCUP dataset is used for detecting attacks.
- The dataset contains information about the duration, flag, service, src bytes, destbytes, and class labels.

Data Preprocessing

- Data preprocessing is the process of removing unwanted data from a dataset.
- Missing data removal
- Encoding Categorical Data
- **Missing data removal**: In this process, null values such as missing values are removed using the imputer library.
- **Encoding** That categorical data is defined as variables with a finite set of label values. Most machine learning algorithms require numerical input and output variables. An integer and one hot encoding are used to convert categorical data to integer data.

Splitting the dataset into train and test data

- Data splitting is the act of partitioning available data into two portions, usually for cross-validation.
- One portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when you separate a data set into a training set and a testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

Feature Extraction

1. Feature scaling Feature scaling is a method used to standardise the range of independent variables or features of data. In data processing, it is also known as data normalization, and is generally performed during the data preprocessing step.

2.Feature Scaling or Standardization: It is a step of data preprocessing which is applied to independent variables or features of data. It basically helps to normalise the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

Classification

SVM:

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships.

Support Vector Machine (SVM) is primarily a classifier method that separates cases of different class labels by constructing hyperplanes in a multidimensional space.SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables, a dummy variable is created with case values of either 0 or 1.

CNN:

Convolution neural networks, like neural networks, are made up of **neurons** with **learnable weights** and **biases.** Each **neuron** receives several **inputs,** calculates a weighted **sum** over them, **passes them** through an **activation function,** and responds with an **output.** The whole network has a **loss function** and all the tips and tricks that we developed for neural networks still apply to convolutional neural networks**.**

Prediction

- It's a process of predicting the attacks on the network from the dataset.
- This project will effectively predict the data from a dataset by enhancing the performance of the overall prediction results.

V. RESULT SIMULATION &DISCUSSION

For network intrusion detection, we use three labeled real training datasets, consisting of benign and malicious network traffic collected in large organizations made up of nearly 10,000 hosts.

Create a label through marking the streams triggered by your company's network IDS as malicious we emphasize these issues during a set of machine learning experiments used to detect network intrusion. The purpose is to demonstrate that the results obtained with the similar ML algorithm are very dissimilar in environments where different numbers of functions or training datasets change.

CNN classifiers to detect network intrusion the final result will be generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like, To assess the classifier's effectiveness, a confusion matrix is required, which provides the number of correct and incorrect predictions based on known true values. True Positive (TP): the actual value is true and so was predicted by the model. True Negative (TN): the actual and predicted values are both false. The actual value is false, but the model predicted it to be true; and the actual value is true, but the model predicted it to be false.

IOU: The Jaccard Index, also known as the overlap metric, is a good evaluation metric for measuring overlap between two bounding boxes or masks in a segmented image. The area of overlap between the predicted segmentation and the ground truth is divided by the area of union between the predicted segmentation and the ground truth to calculate the area of union. This metric has a range of 0–1 (0–100%), with 0 representing no overlap and 1 representing perfectly overlapping segmentation. Using a threshold of 0.5, our goal is to achieve an IOU value of 97% or higher.

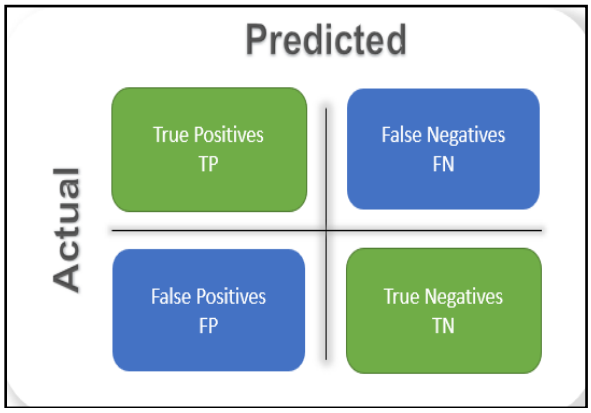


Figure 3 Model evaluation parameters

Green region: Our model estimates 1 (lesion mask), and the ground truth is 1. (True Positive, TP) Blue region: Our model estimates 1 (lesion mask), but the ground truth is 0. (False Positive, FP Our model estimates 0 (absence of a lesion) but the ground truth is 1. (False Negative, FN) Our model estimates 0 (absence of a lesion) and the ground truth is 0. (True Negative, TN)

Accuracy: Accuracy is the measure of how often a model has predicted the right value as per the given input. But it does not give detailed information regarding FP and FN. For some applications where FP and FN are considerable, F1 score and recall play a very important role. Accuracy is calculated by the formula presented in Equation 5.1.

Accuracy=(TP+TN)/(TP+TN+FP+FN) Equation 5.1

Precision: This evaluation parameter tells how frequently a model predicts true positives. The low value of precision infers high false positives. Equation 5.2 presents a formula for calculating precision.

Precision= TP/(TP+FP) Equation 5.2

Recall: This parameter gives information regarding how often a model predicts false negatives. The low recall value indicates that the model predicted a high number of false negatives. Equation 5.3 gives a formula for calculating recall.

Recall= (TP+TN)/(TP+FN) Equation 5.3

2. Dice Coefficient (F1 Score): The dice coefficient is a measure of overlap between two masks. 1 indicates a perfect overlap while 0 indicates no overlap. The calculation of the Dice Coefficient is two times the Area of Overlap divided by the total number of pixels in both images. This metric is correlated to IOU. Our goal is to achieve an F1 score of 95% or better.

F1 Score: The F1 score is calculated by combining both precision and recall. That is, a high F1 score indicates a low number of false positives and false negatives, which infers that the model is accurately detecting actual threats and is not bothered by false alarms. The formula for calculating the F1 score is presented in Equation 5.4.

F1 score=2*(Precision*Recall)/(Presiion+Recall) Equation 5.4

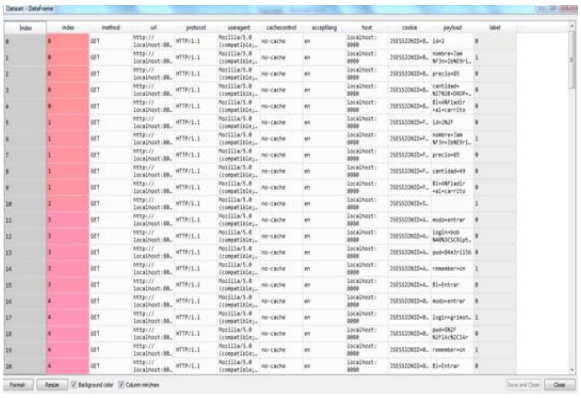


Fig 4 Dataset

The training dataset has a total of 121 "ATTACK" class examples and 1291 examples for the "NORMAL" class. This dataset was collected after the preprocessing phase, and these examples formed the training set. The SVM was trained and summed up the training results where the generalisation error obtained by the classifier is listed under different parametric values.

International Journal of Scientific Research & Engineering Trends
Volume 8, Issue 4, July-Aug-2022, ISSN (Online): 2395-566X

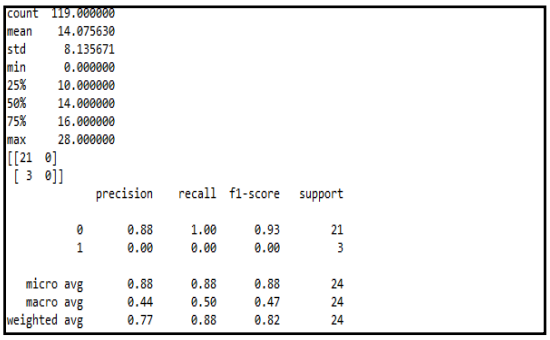


Fig. 5 feature for SVM algorithm.

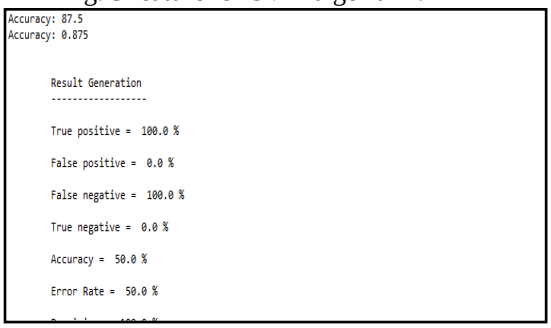


Fig. 6 Parameter for SVM classifier.

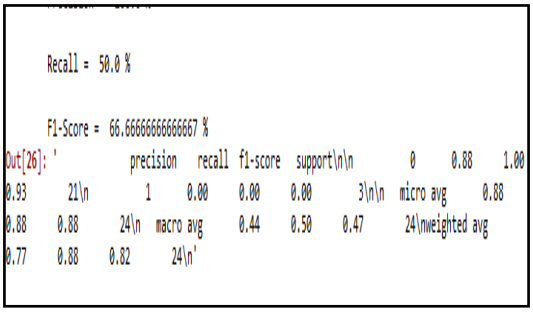
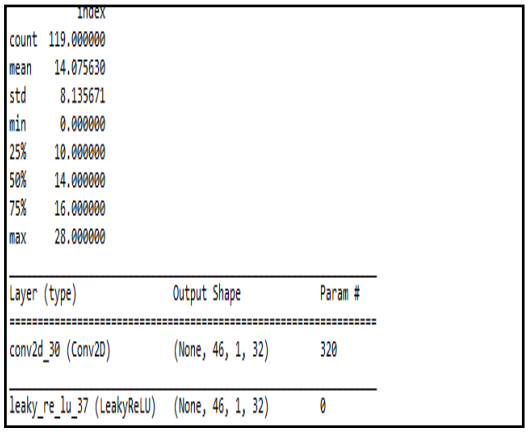


Fig. 7 Features for CNN algorithm



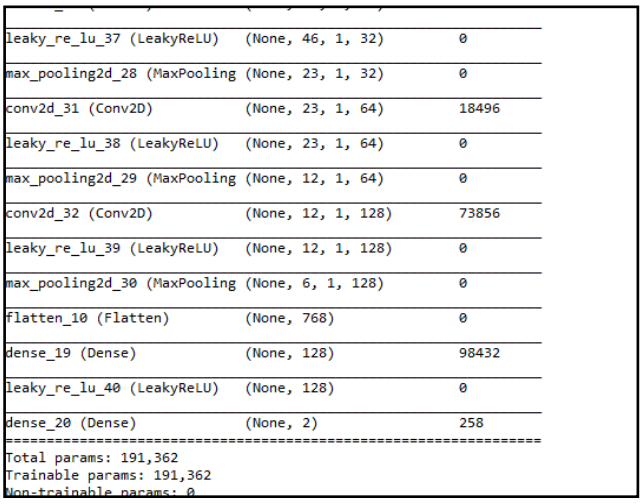


Fig. 8 training data by CNN classifier.

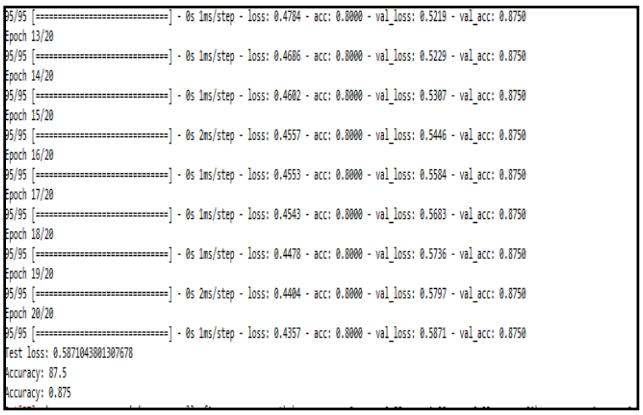


Fig 9 Epoch time for training data

It is clear that these two classifiers achieve comparable detection performance on the data. The situation for intrusion detection is different because modern solutions can achieve higher accuracy results, and while near-perfect accuracy appears to be a significant result, the large number of events occurring each day constitutes thousands of false positives. Self-developed CNN and SVM classifiers trained on the first dataset are shown in Table 1.

Table 1 Performance of the Intrusion Detection Classifier

	Previous Work	Proposed System	
Parameter	ANN	SVM	CNN
F1-score	0.7306	93.00	88.00
Precision	0.7757	88.00	82.00
Recall	0.6270	88.00	50.00
Accuracy	80.12	87.5	87.5

VI. CONCLUSION

The various machine learning algorithms that influence intrusion detection IDS with a high detection speed and a

low false positive rate can be designed using the deep learning capabilities of ML and DL technologies, and the system can quickly adapt to changing malicious behaviour. We categorise these algorithms as artificial intelligence (AI) and computational intelligence (CI) (CI). Although these two types of algorithms have many similarities when dealing with noisy data, some characteristics of CI-based technologies (such as adaptive, fault tolerance, high computational speed, and error recovery features) have demonstrated their necessity. Learning methods are increasingly being used in a wide range of applications, including network security, so it is critical to understand when and which algorithms can achieve adequate results. We examine three related cyber security issues: intrusion detection, malware analysis, and spam detection. The characteristics of ML techniques allow for the design of intrusion detection systems with high detection rates and low false positive rates, while the system quickly adapts to changing malicious behaviour. The characteristics of ML techniques, such as adaptation, fault tolerance, high computational speed, and error resilience in the face of noisy data, confirm the need for efficient intrusion detection systems.

REFERENCE

1. C. Yin et al. Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access, 5, 21954-21961.
2. S.S. Roy et al.A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection. In International Conference on Mathematics and Computing (pp. 44-53). Springer, Singapore, January 2017.
3. Z. Li, et al. Intrusion Detection Using Convolutional Neural Networks for Representation Learning. In International Conference on Neural Information Processing (pp. 858-866). Springer, Cham, November 2017.
4. N. Shone et al. A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50, 2018.
5. U. Fiore et al. Network anomaly detection with the restricted Boltzmann machine. Neurocomputing, 122, 13-23, 2013.
6. G. Hinton et al. A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554, 2006.
7. G. Hinton et al. Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.
8. Y. LeCun et al. Deep learning. nature, 521(7553), 436, 2015.
9. I. Goodfellow et al. Deep learning (Vol. 1). Cambridge: MIT press, 2016.
10. R. Salakhutdinov. Learning deep Boltzmann machines using adaptive MCMC. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 943-950), 2010.
11. P. Vincent et al. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103). ACM, July 2008.
12. H. Debar et al. Towards a taxonomy of intrusion-detection systems. Computer Networks, 31(8), 805-822
13. K. Cho et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation.. In The Conference on Empirical Methods in Natural Language Processing. 1724â€ “ 1734, 2014.
14. X. Li et al. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 4520â€ “ 4524, 2015.
15. [27]X. Glorot et al. Understanding the difficulty of training deep feedforward neural networks. In The 13th International Conference on Artificial Intelligence and Statistics, Vol. 9. JMLR.org, 249â€ “ 256, 2010.