Network-based Intrusion Detection Systems (NIDS)

Aishwarya Shankar

Department of Computer Engineering, San Jose State University

***Abstract*— As the amount of devices purchased and connected to networks continues to increase globally, so do the network security risks. Organizations are increasingly turning to IDSs (intrusion detection systems) to monitor and protect their networks and digital assets. NIDSs (network intrusion detection systems), especially, have become of paramount importance as they are deployed on the entire network and monitor and detect intrusions across the network. Work on NIDSs has spanned a couple of decades and has included approaches like signature-based intrusion detection and anomaly-based intrusion detection using conventional ML (machine learning techniques). These approaches have their advantages and disadvantages; however, in a world where the threat landscape is ever-evolving, researchers are continuing to do further work on NIDSs by looking into more advanced approaches, like using DL (deep learning) for network intrusion detection. In this paper, we delve into the past work done on NIDSs with the aforementioned methods, as well as the newest work being done using deep learning.**

I. Introduction

An IDS (Intrusion Detection System) is software or hardware which has the purpose of detecting abnormal or suspicious activities on a target [1].

The word "intrusion" in security refers to an attempt to compromise any aspect of the CIA triad of a host or network. The CIA triad encompasses the security pillars of Confidentiality, Integrity, and Availability. Intrusions typically occur by way of threat actors, who are individuals or groups, with malicious intent to harm devices or systems.

With the rapidly growing presence of such threat actors globally, as well as the ever-increasing purchase of devices, increased network connectivity, and reliance on information systems, organizations are increasingly utilizing IDSs to monitor and protect their systems [2].

According to [2], there are several key reasons for why organizations utilize (or should utilize) IDSs. Listed below are a few of the most compelling reasons:

1. To detect attacks and security violations
2. To detect preambles to attacks. The can include occurrences like network probes and other such "doorknob-rattling" events
3. To document an organization's existing threats.
4. To act as quality control for security activities. This is typically useful for large organizations.
5. To provide information about any intrusions that occur in the system. This is useful for analyzing the issue and contributing factors, diagnosing it, and taking further actions to remediate it.

A perusal of Gartner, Inc.'s (a leading and respected technology research and consulting firm) insights on the usage of IDSs shows some of the industries and sectors where they are being used most prevalently [3]:

1. Large Enterprises
2. Government
3. Financial Institutions
4. Healthcare
5. Education

It is easy to understand why the above industries would rely on IDSs for their security ecosystems:

- Large enterprises would want to ensure that their digital assets and networks are protected against malicious attacks and data breaches. For government agencies at all levels, it is of paramount importance to protect their infrastructure as well as information about citizens from cyber threats.
- In the financial sector where there are millions of transactions occurring every single day, it is key that the banking platform, customer data, and transactions are kept protected from threats like identity theft and fraud.
- Healthcare providers deal with a lot of sensitive information and PII (personally identifiable information) like patient data and health records.

They need to ensure that these are protected with the utmost security from threats like data breaches.

- Lastly, educational institutions like schools, colleges, and universities need to protect student and faculty data, their academic networks, and administrative systems from being accessed by unauthorized individuals or groups. IDSs help these industries and sectors achieve proactive monitoring and detection of such threats and intrusions.

IDSs can be classified in two major ways - by detection method and by deployment method. There are two categories of detection method-based IDSs - SIDS and AIDS. SIDSs (Signature-based intrusion detection systems) are also known as "knowledge-based" intrusion detection systems and are based on the concept of determining a signature of different attack patterns. AIDSs (Anomaly-based intrusion detection systems) are also known as "behavior-based" intrusion detection systems and are based on the concept of defining a profile for normal activity on a target and measuring subsequent activity relative to this profile. There are two categories of deployment-based IDSs - HIDS and NIDS (each of these can be further classified as SIDS or AIDS based on whether they use signature-based or anomaly-based detection methods as discussed previously). Deployment method has to do with the target, or what the IDS is being used to monitor. An HIDS (Host-based Intrusion Detection System) is deployed on a single host machine and monitors that single host for intrusions or security policy violations. On the other hand, an NIDS (Network-based Intrusion Detection System) is deployed on the entire network itself and monitors the entire network traffic for intrusions and security policy violations [4]. Figure 1 shown below shows the IDS classifications and detection methods within each.

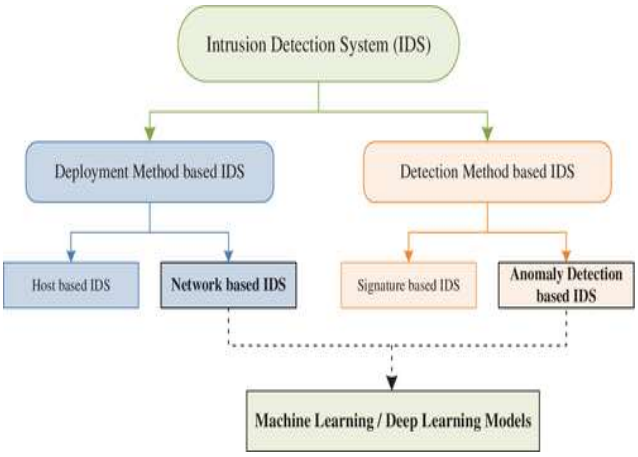

Fig 1. IDS Classifications and Detection Methods

Source: Adapted from [4]

With the rise in cyberattacks globally and to keep pace with new threats, the NIDS especially is a very popular and important monitoring solution deployed by many enterprises to protect their digital assets and networks [5]. In this paper, we delve deeper into the NIDS (network-based intrusion detection system), past work done on it, and recent innovations in this area using deep learning.

II. Previous Work

In this section, we discuss various past work that has been done on NIDSs, beginning with the more simple signature-based intrusion detection method to the more complex anomaly-based intrusion detection methods.

Signature-based intrusion detection is also known as a "knowledge-based" intrusion detection and can be used within an NIDS. The key idea of this method is in performing "pattern-matching", whereby network packets are compared against a database of intrusion signatures. If activity in the network matches an intrusion signature in the database, then it is flagged, and an alarm is triggered. Signature-based detection, as described above, is very useful and accurate for detecting attacks that are already known and whose signatures exist in the database [6]. However, this method is largely ineffective for detecting what are known as "zero-day" attacks. Zero-day attacks are attacks that have not been seen before. Their novelty makes them difficult to detect and defend against [7]. In other words, since these attacks don't have any known signature which is stored in the signature database, signature-based detection is rendered ineffective against zero-day attacks, which is not ideal in current times where targeted attacks and polymorphic malware are on the rise [6].

Anomaly-based intrusion detection in NIDS, on the other hand, has garnered more attention in recent times due to its ability to overcome limitations of signature-based intrusion detection, such as being able to identify zero-day attacks. In anomaly-based intrusion detection, machine learning, statistics-based, or knowledge-based methods are utilized to model the normal baseline behavior of the network. If there is any major deviation detected in the observed behavior in the network compared to the baseline model, it is flagged as an intrusion. The rationale behind this method is that typical user behavior differs from malicious or abnormal behavior. There are two main phases to develop anomaly-based intrusion detection systems: the training phase (where the normal behavior of a network is learned by studying the normal network traffic profile) and the testing phase (where a new dataset is utilized to study the system's ability detect previously unseen intrusions) [6].

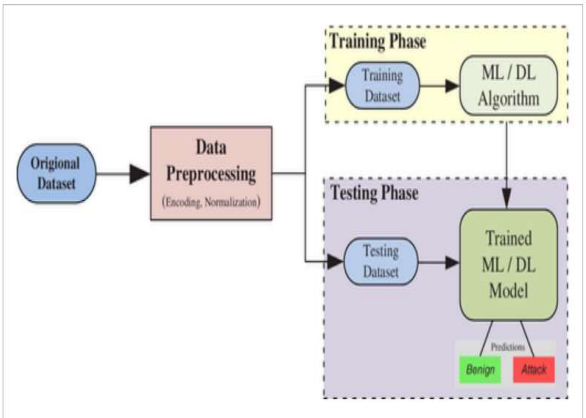

Fig. 2 Phases for Creating an ML Classifier for Network-based Intrusion Detection Systems

Source: Adapted from [4]

Many anomaly-based network intrusion detection systems are built using machine learning, which deals with processing and extracting information and patterns from large amounts of network data. The idea behind using machine learning is that it trains the intrusion detection system to detect patterns and improves intrusion detection accuracy. There are two classes of machine learning methods: supervised learning and unsupervised learning. The difference between the two is that supervised learning utilizes labeled training data for the IDS to detect intrusions, whereas unsupervised learning uses unlabeled data.

In an NIDS using supervised learning, each record of the training data contains a network data source and label (normal or abnormal) to classify that data. Feature selection could also be incorporated to train the algorithm on the relationship between the input data and associated label. The resultant trained classifier can then classify new inputs as being normal or network intrusions. Some commonly used supervised learning classification algorithms/models in network intrusion detection systems are Naïve Bayes and Hidden Markov Models (HMM). Naïve Bayes uses probability formulae to determine the probability that a certain type of attack is occurring in the network based on observed behavior within the network. This algorithm looks at features in network activities and the probabilities of them occurring in normal behavior or network attacks. Naïve Bayes is used widely in NIDS because it is easy to use, and the calculations are efficient. One drawback that was noted with the Naïve Bayes model is that it provides reduced accuracy when the datasets are very large. Hidden Markov Models (HMMs) are models which are trained on various known features of malware. This trained model is then utilized to score incoming network traffic. If the determined score is higher than a predefined threshold value, then this would indicate malware. If the determined score is lower than the predefined threshold, then the network activity would be classified as normal [6]. Similar to Naïve Bayes algorithm, Hidden Markov Models also tend to fail when the quantity of data (in this case, incoming network traffic) is very large. This is referred to as the "curse of dimensionality" [8]. Figure 3 shown below is a chart comparing the performance of various supervised learning algorithms in network intrusion detection from a study conducted by researchers from Hungarian and Jordanian universities. Some insights that can be drawn from this chart are that the Random Forest algorithm performed very well, having a very high TP (true positive) rate of 93.8% and a very high precision value of 99.1%. On the other hand, the Decision Table algorithm performed rather poorly, having a TP rate of 92.4% and the lowest precision value of 94.4% [11].

Machine Learning Classifiers	TP Rate	Precision
J48	0.931	0.989
Random forest	0.938	0.991
Random tree	0.906	0.992
Decision table	0.924	0.944
MLP	0.919	0.978
Naive Bayes	0.912	0.988
Bayes Network	0.907	0.992

Fig. 3 Machine Learning Classifiers Performance for Network Intrusion Detection

Source: Adapted from [11]

In an NIDS using unsupervised learning, the model is trained to identify network intrusions by using unlabeled data. The input data (in this case, the observed network activities) are grouped, or "clustered", into various classes via the learning process. Typically, the normal network activities would form larger clusters, while intrusions would form smaller clusters. In addition, since normal network activity and intrusions are not similar, they would not be grouped into the same cluster. The figure below shows an example of what clustering looks like in network intrusion detection. A popular unsupervised learning algorithm used in many network intrusion detection systems today is K-means. In the K-means technique, 'n' data inputs are grouped into 'k' clusters. Each data input is placed into the cluster which has the nearest mean. K-means has been found to be a good algorithm to use for network intrusion detection because it is simple and efficient. However, one drawback of this technique is that the number of clusters which will be created needs to be pre-defined by the user and can have implications for the

accuracy of the output [9]. Figure 4 shown below is an example cluster diagram for network intrusion detection based on K-means clustering. As can be seen, the normal network activities typically form larger clusters, while intrusions form smaller, isolated clusters.

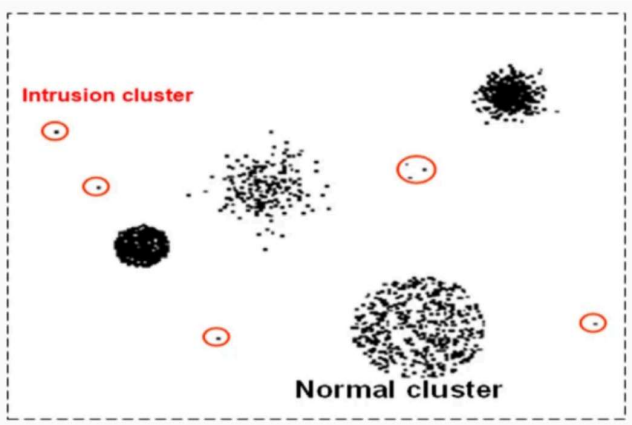

Fig. 4 Example K-Means Clustering Output for Network Intrusion Detection

Source: Adapted from [6]

Also shown below in Figure 5 is an ROC (Receiver Operating Characteristic) graph for K-means clustering from a study conducted by Berhampur University in India. ROC is an important metric to gauge the performance of intrusion detection systems. ROC curves show the tradeoff between the false positive rate and true positive rate of an intrusion detection system. As can be seen, K-means clustering performed moderately well in this study, reaching a true positive rate of over 90%, while having a relatively low false positive rate at around 6% [10].

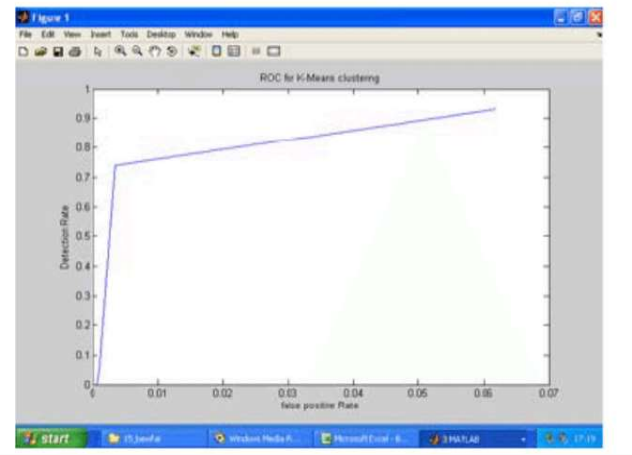

Fig. 5 ROC for K-Means Clustering

Source: Adapted from [10]

III. Problem

While the previous work done on NIDSs is certainly important and useful, the previously discussed approaches certainly have their shortcomings.

With signature-based intrusion detection, it is good for detecting known attacks that have a signature present in the database. However, this method fails for detecting zero-day attacks, which are new and unseen attacks. In current times, where new attacks are coming up all the time, this method would not suffice by itself [12].

With the previously discussed anomaly-based intrusion detection methods, while they are good for detecting zero-day attacks, they can still suffer from a false positive (FP) problem. In addition, conventional machine learning techniques used with this approach typically require manual intervention - humans need to extract features, and this makes it difficult to deploy on large platforms [12].

Hence, a different method is needed in current times to address these shortcomings - deep learning (DL) is one such approach that is gaining a lot of attention in being used for NIDSs.

IV. Technical Approach

DL (Deep Learning) is a subset of ML (Machine Learning), which uses ANNs (artificial neural networks) to mimic how the human brain works. It is self-learning, as it can learn from data, distinguish patterns, and become progressively more intelligent [13].

The neural networks present in deep learning models have hierarchical layers, which find the proper high-level features to use from the raw input data instead of using manually extracted features [12].

There are certain disadvantages to the deep learning approach in NIDSs: it is more compute-intensive and often requires specialized machines to train and test the models, and it also requires more time to be spent in the training phase of creating a model [12]. However, the general consensus at this time is that the advantages of using deep learning for intrusion detection outweigh the disadvantages.

There are several advantages to using deep learning in modern NIDSs: the NIDS is more adaptive and resilient, it provides more detection accuracy, it does not require manual human intervention for feature extraction, and it can detect known and zero-day attacks [14].

Below, we discuss the results of two different studies conducted on the performance of NIDSs using deep learning.

V. Results

Study # 1

The first study, "Network Intrusion Detection Based on Deep Learning", was conducted by researchers from Missouri University of Science and Technology and published in the Procedia Computer Science Journal.

The deep learning model that they developed used CNN (convolutional neural network) and was trained on the UNSW-NB15 dataset. This dataset is popular as it represents real-world network traffic well and accounts for many common vulnerabilities and exposures.

Figure 6 shown below shows the attack categories represented by the UNSW-NB15 dataset.

Table I. Attack categories and descriptions

Attack	Short Description
Normal	Benign network traffic
Fuzzers	Malignant traffic related to spams, penetrations or port scans
Analysis	Attack related to intercepting and or examining network traffic through penetrations or scan
Backdoors	Attack pertaining to use of mechanism designed to bypass security measures
DoS	Attack aimed at flooding network resources making it inaccessible
Exploits	Exploitations through security holes in O.S. or other software applications
Generic	Attacks related to block-cipher, brute force or cryptanalysis
Reconnaissance	The target system is observed for vulnerabilities
Shellcode	Short code 'payloads' to navigate through the system and gain control
Worms	Malicious code that replicates itself to spread through the network

Fig. 6 Attack Categories Represented by UNSW-NB15 Network Traffic Dataset

Source: Adapted from [14]

This model performed very well for network intrusion detection, achieving a good ~94% accuracy in its detection capabilities [14]. Figure 7 shown below, a chart of results from their research, shows the same.

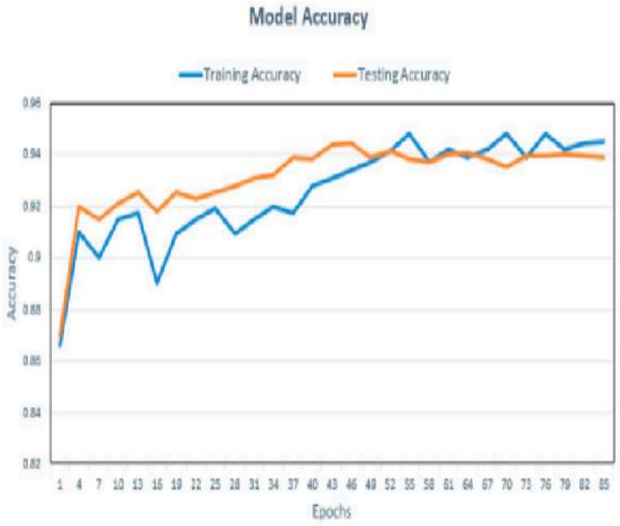

Fig. 7 Deep Learning Model Accuracy for NIDS in Study 1

Source: Adapted from [14]

Study # 2

The second study, "Network Intrusion Detection System using Deep Learning", was conducted by researchers and published as an IEEE Conference publication.

The deep learning model that they developed used DBN (Deep Belief Network) and was trained on a very popular network traffic dataset called the KDD CUP'99. This dataset was developed by MIT's Lincoln Laboratory.

As can be seen below in Figure 8 below, the model achieved nearly 96% accuracy in its network intrusion detection capabilities (represented by the green line marked DBN), which is very respectable [15]. This is a much higher value than was seen with the previously discussed conventional signature-based and anomaly-based network intrusion detection methods.

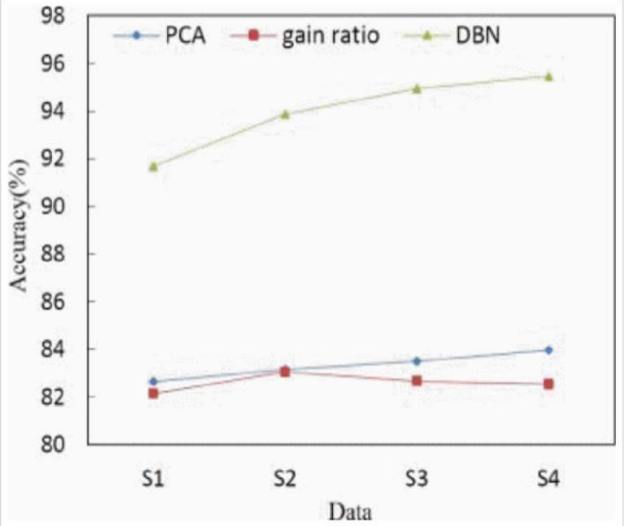

Fig. 8 Deep Learning Model Accuracy for NIDS in Study 2

Source: Adapted from [15]

VI. Discussion and Conclusion

As network security threats and the presence of threat actors continue to rise globally, organizations need to use every tool at their disposal to keep up with and mitigate these threats. Network intrusions detection systems (NIDSs) are very important mechanisms that provide proactive network monitoring and intrusion detection.

The previous work done in the world of NIDSs includes signature-based detection and anomaly-based detection methods. However, as discussed, the traditional approaches in these areas have various shortcomings such as being unable to detect new attacks (signature-based detection), having a false positive problem (anomaly-based detection using ML), and requiring manual human intervention for conventional machine learning techniques (anomaly-based detection using ML) [12].

The most recent work being done in NIDSs uses deep learning (DL) in anomaly-based detection to combat some of the aforementioned shortcomings. DL provides greater accuracy in network intrusion detection, is self-learning, and is more adaptive and resilient to evolving threat landscapes [14]. Using DL for network intrusion detection shows great promise and continues to be an area being actively researched.

References

[1] F. Tchakounté and F. Hayata, “Supervised learning based detection of malware on Android,” *Mobile Security and Privacy*,pp.101–154,2017. doi:10.1016/b978-0-12-804629-6.00006-7

[2] R. Bace and P. Mell, “Intrusion Detection Systems,” National Institute of Standards and Technology, http://cs.uccs.edu/~cchow/pub/ids/NISTsp800-31.pdf.

[3] “Best intrusion detection and prevention systems reviews 2024 | gartner peer insights,” Gartner Inc., https://www.gartner.com/reviews/market/intrusion-prevention-systems (accessed May 13, 2024).

[4] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and Deep Learning Approaches,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, Oct. 2020. doi:10.1002/ett.4150

[5] “What is an intrusion detection system?,” Palo Alto Networks, https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids (accessed May 13, 2024).

[6] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: Techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, Jul. 2019. doi:10.1186/s42400-019-0038-7

[7] F. Deldar and M. Abadi, “Deep learning for Zero-day malware detection and classification: A survey,” *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–37, Sep. 2023. doi:10.1145/3605775

[8] W. Zegeye, R. Dean, and F. Moazzami, “Multi-layer hidden Markov model based Intrusion Detection System,” *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 265–286, Dec. 2018. doi:10.3390/make1010017

[9] B. Bohara, J. Bhuyan, F. Wu, and J. Ding, “A survey on the use of data clustering for intrusion detection system in cybersecurity,” *International Journal of Network Security & Its Applications*, vol. 12, no. 1, pp. 1–18, Jan. 2020. doi:10.5121/ijnsa.2020.12101

[10] M. Panda and M. R. Patra, “SOME CLUSTERING ALGORITHMS TO ENHANCE THE PERFORMANCE OF THE NETWORK INTRUSION DETECTION SYSTEM,” *Journal of Theoretical and Applied Information Technology* , Aug. 2008.

[11] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, “Evaluation of machine learning algorithms for Intrusion Detection System,” *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, Sep. 2017. doi:10.1109/sisy.2017.8080566

[12] J. Lansky *et al.*, “Deep learning-based Intrusion Detection Systems: A systematic review,” *IEEE Access*, vol. 9, pp. 101574–101599, 2021. doi:10.1109/access.2021.3097247

[13] A. S. Gillis, E. Burns, and K. Brush, “What is deep learning and how does it work?,” TechTarget, https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network (accessed May 13, 2024).

[14] L. Ashiku and C. Dagli, “Network Intrusion Detection System using Deep Learning,” *Procedia Computer Science*, vol. 185, pp. 239–247, 2021. doi:10.1016/j.procs.2021.05.025

[15] W. Peng, X. Kong, G. Peng, X. Li, and Z. Wang, “Network Intrusion Detection Based on Deep Learning,” *International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2019