A Review on Deep-Learning Based Network Intrusion Detection Systems

Saket S. Jajoo[1] and Kakelli Anil Kumar[2]
(Corresponding author: Saket S. Jajoo)
School of Computer Science and Engineering, Vellore Institute of Technology[1]
Vellore, TN 632014, India
Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology[2]
Vellore, TN 632014, India
Email: saketjajoo77@gmail.com
(Received Sept. 10, 2021; Revised and Accepted Nov. 15, 2021; First Online Nov. 16, 2021)

Abstract

Network Security is an extremely arising field that secures frameworks, organizations, and information from advanced attacks. With the evolution of the Internet and the development of different types of cyberattacks, developing advanced cybersecurity tools has become an important task to protect any potential breach of sensitive data. This paper introduces the concept of Network Intrusion Detection Systems, elucidates their different types, and compares how different techniques of Deep Learning are implemented in the Network Intrusion Detection Systems to detect any malicious breach in a network. Furthermore, the paper compares the use of Machine Learning and the use of Deep Learning methods and which approach is more effective to detect unwanted network intrusions. This paper further reviews the literature about Network Intrusion Detection Systems and the domain of Machine Learning and Deep Learning that could help make informed decisions about such network intrusions and aid in their prevention and mitigation.

Keywords: Artificial Neural Networks; Deep Learning; Generative Adversarial Networks; Network Intrusion Detection; Recurrent Neural Networks

1 Introduction

In this ever-evolving world, advancements in technology are imperative. We are much more connected today than we ever were in the past. Almost everyone now has instant access to the required data, and can even share it with anyone on the internet. While this trading of data is usually trivial and insignificant, some instances of the activity attract malicious people yearning to gain potentially sensitive information about others. Such malicious people (i.e. hackers, attackers) try to intercept communique, using different approaches, to obtain sensitive information. Though, there are some basic prevention measures, like the Transport Layer Security (TLS) that prevents an outsider to listen in and understand the network traffic packets in the communication channel (as they are encrypted), what if the outsider gets inside a network thus eliminating the need to decrypting the TLS encryption all together? This is where concepts like Network Intrusion Detection Systems (NIDS) and Network Intrusion Prevention Systems (NIPS) come into play.

The main aim of NIDS is to monitor and notify any suspicious activity in the part of the network where it is deployed, and the goal of NIPS is to activate the defenses and prevent the outsider from accessing any further information or penetrating any further deep inside the network. Once NIDS and NIPS learn about traffic behavior and patterns, they can take necessary actions to prevent an outsider from even becoming a part of the network. This certainly enhances the security posture of the network. The primary function of a NIDS is to notice, assess, monitor and report/log the network traffic using predefined methods and techniques to separate ordinary and malicious traffic. Timely detection and appropriate actions can prevent serious consequences like a botnet attack, a brute-force attack, network reconnaissance, DoS/DDoS attacks, network infiltration, account takeovers, etc. There are primarily 2 methods [13] that form the basis of how a NIDS detects and identifies an intrusion event: based on the signature of a network packet or based on anomalous events. The signature-based systems mainly focus on searching for a particular signature (or a previously unknown pattern) within an event. Majority of the NIDS fall under this category.

The primary overhead of this system is that the database of signatures must be kept updated at all times. This implies that signature-based NIDS are just as good as the state of their corresponding 'signature database'. Thus, attackers can easily circumvent signature-based detection by different techniques like obfuscating the network packets using encryption, encoding or lossless compression methods, session slicing and fragmentation, etc. The other type of NIDS (anomaly-based) searches for the sorts of obscure attacks that signature-based IDS finds hard to recognize. Because of the rapid development in malware and attack types, anomaly-based IDS utilizes the concept of machine learning to deal with the detection of intrusions. Anomaly-based IDS looks at the network traffic behavior instead of the actual malicious payload and thus can be evaded if the attacker can make the payload packet coherent with all of the other network traffic. Typically, attackers need a rough layout of the network architecture, devices involved and the protocols used. Thus, they initially probe the network to gather such general information before launching an attack. This unwanted probing can be prevented by this type of IDS as it is capable of identifying and detecting any unwanted network reconnaissance or sweeps.

Machine Learning is a concept in Computer Science that can allow machines 'intelligent decisions' concerning an event or a process with minimal human intervention. There are various algorithms of Machine Learning that are used in anomaly-based NIDS to identify suspicious activities. For instance, a decision tree-based random forest algorithm or Support Vector Machine algorithm can be employed to categorize and classify a network packet as either malicious or benign. Going along the same lines further comes the concept of Deep Learning. The idea here is to make the computers mimic how a human brain works. Similar to the brain structure in humans, deep learning has neurons associated with activation functions that either activate (give a positive result) when it detects a signal beyond a certain threshold. Using the base as Mathematics (calculus and statistics), Deep Learning can represent functions of higher dimensions and complexities which can thus help in increasing the accuracy with which a machine derives an answer to the given task. Thus, deep learning is much more powerful as compared to Machine Learning. Machine Learning requires pre-processing of data (like cleansing, removing null values, etc.) for the algorithm to give accurate results. But in the case of deep learning, the raw data can just be given as input and it can be processed with the help of deep-learning frameworks itself before analyzing it to give accurate predictions and results. This article is focused on reviewing various approaches of how the techniques of deep learning have been employed in NIDS to detect certain events and how accurate the methods are.

2 Literature Review

There have been numerous papers published depicting the efforts connecting Deep Learning methodologies to the domain of Information Security (particularly the Intrusion Detection and Prevention

Systems). These papers have articulated and discussed various methodologies that have been used in NIDS and the current trends as well as future scope as to how the system can improve further. The main goal of this article is to critically analyze such papers and provide a comprehensive review of the said techniques.

Rony Chowdhury Ripan *et al.* [15] present a Forest Learning-based outlier detection system for classifying outliers in data. In this technique, the authors have used the classification-based algorithms in Machine Learning to identify outlier(s) in a Kaggle-based dataset. The dataset is a typical network packet data consisting of features like protocol, service, flag, data size, and service. In their approach, the authors have prepared the data to be trained by different algorithms to compare their accuracies against the data containing outliers and without. To prepare the data, the raw features have been normalized, encoded and correspondingly scaled. After the initial processing, important attributes have been selected using Recursive Feature Elimination (RFE) [17] to reduce the errors during training. RFE finalizes the features to be used based on recursively considering a smaller subset of features. After selecting the features to be used in training, the isolated-forest approach is used to remove the outliers from training data. Isolation Forest [11] builds an ensemble of "Isolation-Trees" based on a randomly selected value from a feature column and then recursively grouping the leaf nodes having values smaller than the predetermined value to the left of the root node, and the nodes having values greater than the predetermined value to the right of the root node. Outliers in the data are the nodes in the tree whose path length is shorter than the average path lengths of all nodes in the Isolation-Tree. The outlier score of a tree node is calculated using the formula:

$$c(m) = 2H(m-1) - \frac{2(m-1)}{m} \tag{1}$$

wherein m is the size of the sample set and $H(x)$ is the harmonic number based on the Euler-Mascheroni constant, i.e.

$$H(i) = ln(i) + 0.5772156649 \tag{2}$$

Further,

$$s(x,m) = 2^{\frac{-E(h(x))}{c(m)}} \tag{3}$$

where $E(h(x))$ is the average value of $h(x)$ from a collection of Isolation-Trees. The closer the value of s is to 1, the higher the chance of the node being an outlier is. After the outliers are detected, they are removed from the dataset and then are fed to 5 different classification algorithms to classify the Intrusion Data as malicious or benign. The accuracy of an algorithm is calculated using the F1 score which is the harmonic mean of precision and recall. Precision is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. The result of the method is that the accuracy is improved upon removing the outliers from the original dataset.

Though the results are quite promising, there are a few drawbacks to the said approach. The way RFE works is by using several different combinations of different parameters and attributes and then selecting the best set of features for training the model based upon intermediate observations. This makes the process exponentially compute-intensive as more attributes get added to the dataset. Further, isolation trees may not be the best technique used in anomaly detection. The main problem with the isolation trees is the way the trees are branched which can introduce a bias, which is thus likely to reduce the reliability of the anomaly scores for ranking the subsamples. What instead could be used is the Extended isolation forest approach, as discussed by Hariri *et al.* [6], wherein instead of a random selection of feature values within the dataset range, random slopes are instead selected as the decision boundaries for classifying data.

In the paper titled 'Image Classifiers for Network Intrusions' [14], Noever *et al.* proposed a unique and unusual approach towards identifying network intrusions. This approach includes the use of image

classifiers to detect malicious network intrusions. The idea here is to use the UNSW-NB15 attack dataset [13] that consists of raw network packet data, generated features on labeled attacks, and scored statistical methods for identifying each attack family. The approach here is to map the scaled numerical features to images and further use the CNN on them from training, validation, and testing. One advantage of using this method is that incremental training can be used on CNN to train upon new data without training the entire neural network from scratch again. This thus saves a lot of training time and also does not impact the accuracy significantly.

The authors have converted all of the tabular features into images to solve for 9 different attack types (consisting of Analysis, Backdoors, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcodes, and Worms). Once the feature sets are converted to 256×256-pixel images, all the categorical features are one hot encoded and all the numerical features are rescaled between 0-255. The column labels are also included in the training and testing sets as CSVs to compare image-based classification methods to various other statistical approaches like decision trees, random forest, and SVMs. To train the convolutional neural network on this modified data, the train test split is left intact as it was in the UNSW-NB15 attack dataset with a ratio of $1:2$ which thus makes the number of training examples as 82332 and the number of testing records as 175341. To further test the method of transfer learning, the pretrained MobileNetV2 model is used by the authors after altering the final output layer of the neural network. This approach results in a 97 percent accuracy in detecting the attack scenarios and 98 percent accuracy in detecting benign cases.

The results from this approach look promising in detecting 9 different attack classes. Furthermore, since the classifier is trained using the MobileNetV2 model [16], it can easily be used on mobile/edge devices [10] and other smaller electronic appliances. Thus, this approach successfully demonstrates the use of image classifiers on image data, obtained after converting the tabular data into image thumbnails. Though the results are good and the approach is unique, there is a shortcoming to the use of such an approach. The CNNs are built to identify close neighbours and represent them as a unique single entity, while in the case of network attack data, this may not always be true. There may not be a relationship between close data points while the model could categorize them into a particular attack type. One way this problem can be tackled is to shuffle the tabular data and then convert them into images so that the classifier does not correlate and map the nearby data points as 1 entity.

The paper by Boxiang Dong *et al.* [2] which is titled 'Cyber Intrusion Detection by Using Deep Neural Networks with Attack-sharing Loss', the authors proposed another Deep Learning based approach to solve the problem of efficiency of the Deep Learning models in their application in NIDS. The main problem the authors are trying to solve is to build a model that can perform efficiently on a dataset with the following properties:

- The classes of attack vectors are diverse, and,
- The dataset is highly imbalanced (almost 95 percent of records are benign whereas only 5 percent of them represent intrusion events).

Training a model on such a dataset could result in an extremely inefficient model with a high bias towards non-malicious data events. Thus, the authors have architected a new system called DeepIDEA which can detect network intrusion events and classify them accordingly. The main aim behind DeepIDEA is to achieve high detection accuracy on an imbalanced dataset using an attack-sharing loss function which can move the decision curves towards the attack classes and thus reduce the bias towards the majority classes (which are typically benign). The working of DeepIDEA is similar to that of the reinforcement learning approach i.e. the intrusion misclassifications (false positives and false negatives) get a higher penalty as compared to the attack misclassifications. The DeepIDEA model is a fully connected neural network wherein the ReLU activation function is used for all neurons except the ones in the output layer which uses softmax as its activation function to classify the data records. This is a typical neural network

model with a slightly different loss function which makes it effective against imbalanced datasets. A typical neural network generally uses a cross-entropy equation as its loss function which may not be quite accurate in this case as it is unable to consider the misclassifications which thus implies that a penalty cannot be imposed upon such an event. Thus, the authors proposed a modification to the existing cross-entropy function to be implemented in DeepIDEA. The authors have termed their loss function as attack-sharing loss function which has an extra regularization parameter that penalizes mis-classification, the equation to which is defined as:

$$J_{AS} = J_{CE} - \frac{1}{N}\sum_{i=1}^{N}\lambda(\mathbf{I}(y^{(i)},1))\log p_1^{(i)} + \sum_{j=2}^{c}\mathbf{I}(y^{(i)},j)\log\left(1-p_1^{(i)}\right) \tag{4}$$

wherein is the control parameter, J_{CE} is the cross-entropy loss, $y(i)$ is the incident value (1 or 0 corresponding to benign or malicious respectively), $p_j^{(i)}$ is the probability function for a data point (x(i), y(i)) that it belongs to the j^{th} class, and $\mathbf{I}$ is the indicator function defined by:

$$I(a,b) = \begin{cases} 1 & when \;\; a = b \\ 0 & otherwise \end{cases} \tag{5}$$

When is small, J_{AS} behaves like the typical cross-entropy loss function, and when is large, J_{AS} transforms to an objective function to address the binary classification problem (malicious record vs benign record). This λ term term eliminates the bias towards the majority class by shifting the decision curve towards the attack classes. Furthermore, instead of the Stochastic Gradient Descent (SGD) as the optimizer, DeepIDEA employs the Adam as its optimizer which can adaptively update the learning rate and does not suffer from slow asymptotic convergence. In this use of Adam optimizer, the past gradient and its squared are stored in two different variables s and r, also known as the first moment and the second moment. After every mini-batch training, s and r values are updated for bias corrections which thus makes the slope smooth and gentle in the parameter space, which thus helps in faster convergence. The authors tested their approach on 3 different datasets: KDD99, CICIDS17, and the CICIDS18 dataset. On all these datasets, DeepIDEA has proved to be more efficient than other MLP-CEs (Multi-Layer Perceptrons with Cross-Entropy as their loss function). However, there are some shortcomings of DeepIDEA, some of which are that it does not focus enough on highly under-represented classes because the attack-sharing loss function only shifts the decision curve towards the attack classes and not towards some specific class. Furthermore, the aforementioned formula does not categorize the different attack types. In order to minimize the JAS cost function, the classifier tends to categorize every attack record as a majority class. This limitation thus makes DeepIDEA suitable for datasets where benign records are in majority and the attack classes are balanced. Thus, all-in-all, DeepIDEA is suitable for a dataset where an imbalance exists between benign class and the attack classes.

Authors Gastón García González *et al.* in their paper 'On the Usage of Generative Models for Network Anomaly Detection in Multivariate Time-Series' [4] suggest the use of Generative Models concept to detect network anomalies using time-series, GANs (generative adversarial networks) [5], and RNNs. The authors proposed a unique model called Net-GAN which detects network intrusion anomalies in time-series data exploiting temporal dependencies via RNNs. The Net-GAN model can understand the underlying distribution of multivariate data thus having a powerful approach to detecting network anomalies in complex environments. The authors also have architected another model called Net-VAE based on variational auto-encoders (VAE).

GANs and VAEs are the two most powerful methodologies in identifying the underlying data distributions. GANs are a unique way of training generative models by framing the problem as a supervised learning task with two sub-models: the generator model that is trained to generate new examples, and the discriminator model that tries to classify examples as either real or fake. The two models are

trained together until the generator model is generating a significant number of plausible examples. In mathematical terms, to learn a generative distribution p_g over the learning data x, the generator builds a mapping from a prior noise distribution p_z to a data space as $G(z)$. The discriminator outputs a single scalar $D(x)$ representing the probability that input x came from real data rather than from p_g. Behind the scenes, Net-GAN uses an LSTM model for both Generators and Discriminators instead of the traditional MPL as LSTMs can comprehend the underlying temporal dependencies in the data which is essential in analysing time-series data records.

The VAE model's architecture comprises typical encoder and decoder functions which form the auto-encoder (AE). Auto-encoders are a type of neural networks which are used to efficiently learn the representations of input data for dimensionality reduction tasks. The encoders are used to reduce the dimensions and the decoders are used to map the reduced encoding latent spaces as close as possible to the original inputs. The Net-VAE model comprises two data alignment and reconstruction layers to process time-series data, a two three-layer feed-forward neural network acting as an encoder and decoder. The model detects using the residual loss function which categorizes a data point as anomalous if the difference between it (x) and its reconstruction $\widehat{x}$ is greater than a certain threshold.

In terms of time-series processing, Net-GAN and Net-VAE both operate through a sliding window of T samples with unit steps. At each step, distance amongst the time-series packets is calculated and declared as an anomaly if the samples seem to be deviated from the baseline by a certain threshold. To avoid false positives, the sample T represents a moving average of data points.

The CPS and SYN-NET datasets are used to evaluate the performance of the Net-GAN model. The Net-GAN model is tested for the detection of botnet traffic, DDoS attacks, port scan traffic, and infiltration instances. RoC curves are plotted to represent the performance of the Net-GAN model and the Net-VAE model which plots the classification metrics for the models at all classification thresholds. The Net-GAN model is able to identify the malicious traffic with a False Positive Rate (FPR) of less than 1 percent. The Net-VAE model can achieve an accuracy of 70 percent with 0 false positives. Though the accuracy is not that impressive, the FPR seems to play a significant role here in order to not mark benign traffic as malicious. The final results show that the models can detect the botnet, infiltration, port scan, and DDoS attack types with an FPR below 1 percent.

The proposed scheme provides 2 advantages: spotting anomalies in an environment with highly unbalanced data (as there are more negative classes as compared to the positive ones [actual anomaly instances]), and training the model in an unsupervised fashion due to the lack of labeled instances. Furthermore, generative models seem to show the potential in detecting network intrusions in complex environments and high dimensional systems such as modern networks where traditional models may fail to capture the fundamental data distributions.

The shortcomings from LSTM based approach are solved in the paper titled 'Intrusion Detection System in Smart Home Network Using Bidirectional LSTM and Convolutional Neural Networks Hybrid Model' [3] by Nelly Elsayed *et al.*. In this paper, the authors have proposed using a hybrid of Bi-LSTM (Bidirectional Long Short term Memory) approach [18] and CNN (Convolutional Neural Network) to be implemented in NIDS to detect malicious or unwanted network intrusions in an IoT based environment. The BiLSTM model aids in the learning process for NIDS while the CNN can accurately extract the data features. In the Bi-LSTM model, 2 hidden layers of the RNN are present in opposite directions to produce an output. Besides improving data flexibility because of bi-directionally faced layers, RNN also increases the reachability of future state inputs from the current ones and thus does not require fixed input before actual training because of the feedback loop through time. The bidirectional structure imbibes the temporal dynamic of the recurrent system as the model is trained in both the feedforward and backward directions. The proposed architecture comprises 11 layers. The initial layer is the batch normalization layer which normalizes each input record before passing it over for further processing. The overall structure of the network is like that of a CNN, with a slight difference being that in between

the Pooling layer and the 1D Convolution layer and also in between 2 1D Convolution layers comes the Bi-LSTM layer, after which the output is flattened and fully connected to a 2 layer dense network before emitting the final output by using a softmax activation function. Upon training and testing the architecture on the IoT Intrusion Dataset [9], a 98.93 percent accuracy was observed with a Precision of 98.20 percent, a Recall of 99.61 percent, and F1 score of 98.90 percent. The architecture is in fact quite interesting as it uses CNN (which is highly efficient in feature selection) to extract relevant features from the dataset and also consider the past records while training for the subsequent ones which increases the efficiency of both training as well as the model itself.

Going further along to detecting unwanted network intrusions in IoT-based environments, Idriss Idrissi *et al.* proposed another method for detecting botnet attacks in their paper titled 'Toward a deep learning-based intrusion detection system for IoT against botnet attacks' [7]. The NIDS solution they plan to implement is proposed to be deployed on a fog node [1, 8], in an IoT environment. Such a deployment gives NIDS the ability to analyze traffic packets in real-time as well as to monitor the traffic both inside and outside of the network. Furthermore, it also enables monitoring of traffic between insider zombie devices which could potentially be targeted by attackers to become a part of the bot network during a botnet attack. The NIDS solution is called BotIDS which is based on deep learning. BotIDS goes through the typical process involved in Deep Learning like data pre-processing and preparation, training, testing and validation, and deployment. The Bot-IoT dataset [12] is used by the authors to train and test their approach. The output of the trained model can be among 11 different classes in which 1 class is the normal or the benign class and the others are one among 10 different attack types (service scanning, OS fingerprinting, TCP DDoS, UDP DDoS, HTTP DDoS, TCP DoS, UDP Dos, HTTP DoS, key logging and data exfiltration). The authors architected 2 different models: CNN-based and RNN-based (Simple RNN, LSTM, and GRU), which were then trained on the dataset. Furthermore, the training was done in different ways according to how the dataset was split for input: in one approach only the 'best-features' were considered as input to the models, whereas in the other, all features (full-features) were processed. Based on the results, it can be inferred that the CNN-based architecture performs better as compared to RNN, LSTM, and GRU with an accuracy of 99.94 percent and prediction time of less than 0.34 ms. Despite having such a viable accuracy rate, the only concern here is the dataset used. For an IoT-based environment, the numbers look promising but the model is only as efficient as the quality of the dataset on which it has been trained upon. The Bot-IoT dataset is not an 'ideal' dataset meaning that the number of records for each class are not in proportion. Thus, the deep learning model's ability to detect network intrusions with high accuracy in real-world scenarios is limited because of such an imbalanced dataset.

3 Comprehensive Comparison

Rony Chowdhury Ripan *et al.* in their paper [15] provide a method for outlier detection using a forest-based approach to classify cyber anomalies. Though this is not directly used for intrusion detection by the authors, it can be used for the said purpose because an intrusion event can also be considered as an outlier for a given set of network traffic. The main goal of the authors here is to provide a way to deal with the outliers that may be present in the dataset so that when a Machine Learning model learns on the data, the results do not get skewed. Further, using the extended isolation forest approach [6], one can reduce the effect of bias present in the dataset as the algorithm selects random slopes instead of feature values as decision boundaries.

Dong, B. *et al.* in their paper [2] also proposed a way of dealing with biased or an unbalanced dataset to train a model for intrusion detection. Instead of re-engineering the data to reduce bias as proposed in 'An Isolation Forest Learning Based Outlier Detection Approach for Effectively Classifying Cyber Anomalies' [15], the authors propose an algorithm called DeepIDEA to reduce the effect of

unbalanced data points using a reinforcement learning-based approach. The loss function in their model is also different than the usual cross-entropy loss function. They have termed it as 'attack-sharing loss function' that tries to eliminate the bias towards the majority/benign class by moving the decision boundary towards the attack feature classes.

Noever *et al.* in their paper [14] proposed a unique image classification-based approach to tackling the problem of network intrusions. They suggest using CNN (Convolutional Neural Network) for image classification after mapping the data points to image space. The main advantage of using CNN is that it can be incrementally trained if a new dataset is to be used for training. This means that only the last layer of the network will need to be trained with the new dataset saving compute power as well as time significantly. Additionally, since CNNs trained with the ImageNet algorithm can also be deployed at edge devices, this approach can further be extended to IoT-based environments to detect unusual network intrusions before the traffic even enters the backbone network.

Going along the same lines of managing traffic in an IoT-based domain, authors Idriss Idrissi *et al.* in their paper [7] intend to use the general CNN model to classify incoming traffic as either malicious or benign. The only difference here is that they are devising a model which can be run in a decentralized, FoG-based scenario wherein the edge devices do most of the crucial processing of traffic and data. The main goal of their model BotIDS is to identify any botnet attack-related network traffic and issue alerts. By deploying this model to devices that are within the network, they aim to get maximum coverage of all of the network traffic for analysis. Such a deployment not only analyses traffic in real-time but also keeps an eye for any insider zombie devices that could potentially be used by an adversary as a part of the botnet in an attack.

Authors Elsayed, N., Zaghloul *et al.* in their paper [3] also try to cover the IoT-based environment wherein they proposed the use of CNNs and Bi-LSTMs to first extract the relevant features from the data traffic and then detect malicious or unwanted network intrusions. This approach differs from the one mentioned in 'Image Classifiers for Network Intrusions' [14] as instead of using image space for analysis, the authors use time-space (temporal relationships between datapoints) to analyze the incoming network traffic and classify them as either malicious or benign. Bi-LSTM has RNNs placed in opposite directions that increase the reachability of future state inputs from the current ones and thus do not require fixed input before actual training because of the feedback loop through time. Training using only relevant features and keeping a track of temporal relationships among data points increases the efficacy of the model.

Authors Gastón García González in their paper [4] proposed 2 different models Net-GAN and Net-VAE to identify network intrusions even after being trained on an imbalanced dataset. The authors suggest the use of GANs and RNNs for intrusion detection. The main idea here is to identify the temporal relationship between data traffic using RNNs and thus identify the incoming data traffic as malicious or benign This model seems to be made up of all of the positive aspects of models in [2] and [3]. While the Net-GAN model is able to understand the underlying distribution of multivariate data which is a powerful approach to detecting network anomalies as they use LSTM under the hood, the VAE model is able to map higher dimensional data to lower dimensions and as close to the original data point as possible. This reduction in dimension implies a reduction in training time as well as guarantees an inexpensive compute. As compared to the model in 'Intrusion Detection System in Smart Home Network Using Bidirectional LSTM and Convolutional Neural Networks Hybrid Model' [3], which just keeps a track of temporal relationships, this model can be used to spot anomalous network traffic even in complex conditions as they can address the problem of dimensionality reductions besides keeping the track of data packets in the temporal domain. Also, the ability of this model to identify the underlying data distributions aids in improving its accuracy.

4 Conclusion

With the advancements in the area of deep learning, it has been possible to detect network intrusions with good accuracy before any tangible damage is done by the attacker(s). Yet, these 'intelligent' NID Systems have a great potential to improve their detection capabilities based on various factors like how they are architected, what dataset is used to train them, the environment they are deployed in, the part of the network of subnet they guard and so on. Further, as newer technologies develop (like for instance graph-based databases, data structures), NIDS will have to be improved upon to accommodate such developments. As the technological stack for a network and its hosted application grows, the attack surface grows and the attackers will find new attack vectors to try and raid such systems. Thus, the current NIDS will have to cope up with ever-evolving environments to reduce the rate of intrusions and prevent attackers from gaining unwanted access to the networks.

References

[1] H. Chen, X. Jia, and H. Li, "A brief introduction to IoT gateway," in *IET International Conference on Communication Technology and Application (ICCTA'11)*, pp. 610–613, 01 2011.

[2] B. Dong, H. Wang, A. S. Varde, D. Li, B. K. Samanthula, W. Sun, and L. Zhao, "Cyber intrusion detection by using deep neural networks with attack-sharing loss," *ArXiv*, vol. abs/2103.09713, 2021.

[3] N. Elsayed, Z. S. Zaghloul, S. W. Azumah, and C. Li, "Intrusion detection system in smart home network using bidirectional lstm and convolutional neural networks hybrid model," *ArXiv*, vol. abs/2105.12096, 2021.

[4] G. González, P. Casas, A. Fernández, and G. Gómez, "On the usage of generative models for network anomaly detection in multivariate time-series," *ACM SIGMETRICS Performance Evaluation Review*, vol. 48, pp. 49–52, 2021.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14)*, pp. 2672–2680, 2014.

[6] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, pp. 1479–1489, 2021.

[7] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. Elfadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, pp. 110–120, 2021.

[8] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. S. Goren, and C. Mahmoudi, "Fog computing conceptual model," *Special Publication (NIST SP), National Institute of Standards and Technology*, 2018.

[9] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, *Iot Network Intrusion Dataset*, 2019. (`https://ieee-dataport.org/open-access/iot-network-intrusion-dataset`)

[10] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 2017.

[11] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[12] N. Moustafa, *The Bot-IoT Dataset*, Oct. 16, 2019. (`https://dx.doi.org/10.21227/r7v2-x988`)

[13] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Military Communications and Information Systems Conference (MilCIS'15)*, pp. 1–6, 2015.

[14] D. A. Noever and S. E. Miller Noever, "Image classifiers for network intrusions,", 2021. *ArXiv*, vol. abs/2103.07765, 2021.
[15] R. C. Ripan, I. H. Sarker, M. Anwar, H. Furhad, F. Rahat, M. M. Hoque, and M. Sarfraz, "An isolation forest learning based outlier detection approach for effectively classifying cyber anomalies," *Arxiv*, vol. abs/2101.03141, 2021.
[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.
[17] I. Sarker, Y. Abushark, F. Alsolami, and A. Khan, "IntruDTree: A machine learning-based cyber security intrusion detection model," *Symmetry*, vol. 12, no. 5, pp. 754, 2020.
[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.

Biography

Saket S. Jajoo biography. He received his Bachelor in Technology (B.Tech) degree in Computer Science and Engineering (with specialization in Information Security) from Vellore Institute of Technology, Vellore, Tamil Nadu, India in 2020. His research interests include Network Security, Computer Vision, and Cryptography.

Kakelli Anil Kumar biography. Dr. Kakelli Anil Kumar is currently an Associate Professor with the School of Computer Science and Engineering at Vellore Institute of Technology, Vellore, Tamil Nadu, India. He earned his Ph.D., in Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTUH), Hyderabad in 2017. His research interests include Wireless Sensor Networks, Internet of Things (IoT), Cloud Computing, Network Security, Malware Analysis, and Blockchain and Cryptocurrency.