A Comparative Study of Deep Learning Models for Network Intrusion Detection

Jyoti Khurana
Assistant Professor
Information Technology Department
Maharaja Surajmal Institute of Technology, Delhi, India

Vachali Aggarwal
Student
Information Technology Department
Maharaja Surajmal Institute of Technology, Delhi, India

Harjinder Singh
Student
Information Technology Department
Maharaja Surajmal Institute of Technology, Delhi, India

ABSTRACT
With the advancement of digital technologies, cybersecurity is attracting more attention as cyber-attacks are becoming more frequent and threatening. A marked upturn has been noticed in the volume and creativity of hacks and cyberattacks. Artificial Intelligence (AI) and Deep Learning (DL) can help address these concerns by contributing to threat detection. They can recognize patterns in data, enabling security systems to learn from former experience. This paper concerns the comparative evaluation of the several techniques of deep learning employed for network intrusion detection.

Keywords
Cybersecurity, Deep Learning, Deep Neural Networks, Intrusion Detection, Anomaly Detection, Network Intrusion Detection

1. INTRODUCTION
Gone are the days when tasks were performed manually. This is the era of the internet. From buying milk to selling home furniture, online classes to office meetings, banking work, and healthcare systems, technology have provided means to carry out every little thing through the internet.

With the substantial increase in internet dependence and devices [9], the challenge of keeping the devices and systems safe and protected from vulnerability attacks and hacks also increases. These attacks can cause losses in millions of dollars and loss of systems and devices at crucial times and affect the physiological state of parties involved. Recent statistics have shown a surge in information loss in the devices that are common in the workplace, including mobile phones and IoT devices. Neoteric security research suggests that most companies have poor cybersecurity practices and unprotected data, making individuals and companies prone to cyber-attacks. Professional groups and individuals must make cybersecurity best practices and awareness a part of their culture to successfully fight against malicious intent.

The accelerated increase in the attacks raises staggering security challenges. Thus to recognize network attacks and stop them from causing any damage to the network, various techniques including machine learning and shallow learning approaches such as Naive Bayes [10], Decision Trees [11], and Support Vector Machines (SVM) [12] were introduced and implemented. Though these techniques have good detection accuracy, but these also require a high level of human expert interaction and are error-prone in addition to being labor-intensive and process expensive. Furthermore, colossal training data is necessary for proper training and operation, often stumping in a heterogeneous and dynamic environment. With the motive to overcome these limitations, deep learning was introduced for intrusion detection systems. Deep learning has exemplified the better or at least matched shallow learning techniques' performance until now. It facilitates more in-depth analysis of network data and swift identification of any anomalies in network traffic data in real-time.

This survey paper confers a comparative study of deep learning models used for network intrusion detection. This paper highlights various deep learning techniques used in a supervised manner in already published models such as LuNet [2], Pelican [3], Dual Net [13], and Spiking Neural Networks with single spike temporal coded neurons [14] for NIDS. LuNet emphasizes on the hierarchy of combined CNN and RNN layers to extract both spatial and temporal features. Pelican uses the working principle of LuNet and combines it with Residual Learning [26]. Dual Net also uses CNN and RNN but utilizes the self-attention mechanism for feature learning for a better selection of features after the feature extraction stage.

The unsupervised approaches used in Kitsune [4] and AnomalyDAE [5] models were also discussed, which were published earlier as well. Kitsune is a lightweight, plug-and-play intrusion detection system that uses an ensemble model of autoencoders. On the other hand, AnomalyDAE incorporates a self-attention mechanism while using two autoencoders, one for node embedding and another for attribute embedding. Detail discussions are suggested in Section 4.

The remnant of this paper is organized as follows:

Section 2 put forward the Intrusion Detection System and its classification, highlighting the NIDS (Network Intrusion Detection System). In Section 3, the different types of learning techniques are being talked about. Section 4 discusses the several Deep Learning-based models for NIDS (Network Intrusion Detection System) already published and their comparative analysis. Section 5 sheds light on the comparative analysis and finally, Section 6 provides the conclusion.

Table 1. List of Abbreviations

ABBREVIATION	MEANING
DL	Deep Learning
SVM	Support Vector Machines
IDS	Intrusion Detection Systems
NIDS	Network-based Intrusion Detection System

HIDS	Host-based Intrusion Detection Systems
DoS	Denial of Service
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
AccV	Validation Accuracy
AccT	Testing Accuracy
DR	Detection Rate
FPR	False Positive Rate
ML	Machine Learning
DSC	Depth wise separable CNN
GRU	Gated Recurrent Unit
FAR	False Alarm Rate
SSDP	Simple Service Discovery Protocol
OS	Operating System
SYN DOS	Synchronize Denial-of-Service
SSL	Secure Sockets Layer
SNN	spiking neural networks
ANN	Artificial Neural Networks
STDP	Spike-timing-dependent plasticity
KNN	k-Nearest Neighbours

2. OVERVIEW OF IDS

Intrusion Detection Systems or IDS are used to recognize unusual access and detect attacks on the network. It is a software application or a hardware appliance that inspects systems/networks for suspicious activity and policy violations and then issues alert to the system administrator whenever a harmful activity or policy breach is suspected. The most prevalent categorizations of IDS are NIDS (Network-based Intrusion Detection Systems) and HIDS (Host-based Intrusion Detection Systems), as shown in Figure 1.

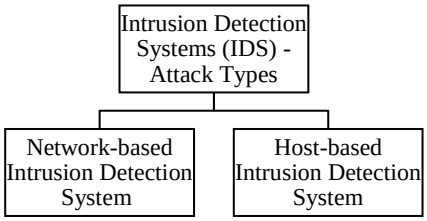

Figure 1. IDS Attack types

While NIDS (Network-based Intrusion Detection Systems) analyzes the network traffic, HIDS (Host-based Intrusion Detection Systems) inspects the internals of a system/server along with the network packets. NIDS works in real-time, wherever HIDS examines historical data to apprehend non-conventional techniques of hacking that might not be easy to detect in real-time.

A NIDS or Network-based Intrusion Detection System is hardware or software-based system or device distributed across the network to inspect the traffic crossing through the systems on which it is installed passively. It detects network born cyber-attacks such as malware/virus replication, DoS (Denial of Service) attacks, and intrusion within the system from internal as well as external sources.

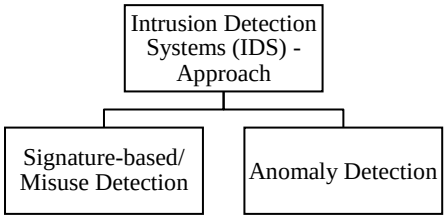

Figure 2. IDS Approaches

NIDS are either based on misuse detection (signature-based) or based on anomaly detection, as shown in Figure 2. Both functions to detect suspicious activity to and from hosts and within traffic itself. Signature-based or misuse-based network intrusion detection systems are very effective against known attacks. They look for behavior with known patterns of events specific to recognized threats in system calls and network traffic. Anomaly-based network intrusion detection systems detect both misuse and intrusion by comparing network activity against the normalized baseline on which it is trained. It monitors network traffic and classifies it as regular or malignant based on the rules or heuristics rather than known patterns or signatures. Signature-based network intrusion detection systems have a few shortcomings, such as swift network data overload, encryption issues, lag time issues with signature development, and inability to identify new or previously not known cyber-attacks while anomaly-based intrusion detection systems concern about the high false-positive rate and improper detection of attacks that attempt to blend in.

3. LEARNING TECHNIQUES

3.1 Supervised Learning

Supervised learning is a typical machine learning / deep learning technique. In this type of learning, a set of examples (known as a dataset) is fed to the model with the label of each example. The purpose of the model is to anticipate the label when input is given to the model. For e.g.:- considering a binary classification problem, a set of input {(x1,y1),(x2,y2),(x3,y3).....(xn,yn)} is given along with its labels {0,1,1,0,1,.....}, where 0 represents a class and 1 represents another class. In this way, the model learns which feature pair belongs to which class so that when a new input is given, the class it belongs to can be predicted by the model.

3.2 Unsupervised Learning

Unlike supervised learning, the data is not fed with the labels in unsupervised learning. Instead, model classifies the data depending upon the typical characteristics of the input data. It automatically organizes the data, searches for common characteristics, and classifies it based on internal knowledge.

In deep learning, autoencoders are considered a type of unsupervised algorithms. But these are supervised algorithms that are trained in an unsupervised manner. Autoencoders try to mimic the input provided to them and are often called unsupervised because, in it, the data is fed without labels; input is itself the label. If hWb(x) is a function of an autoencoder where x is input, then hWb(x)≈x, which means that an autoencoder's output is approximately equal to the input itself.

Table 2. Comparison of supervised and unsupervised learning

	Supervised Learning	Unsupervised Learning
Dataset	Labeled dataset	Unlabeled dataset
Approach	Regression and Classification	Clustering and Association
Method	Bayesian Logic, Linear Regression, Decision Tree, Multi-class Classification, etc.	Apriori Algorithm, KNN, and Clustering, etc.
Computational Complexity	High	Low
Data Analysis	Off-line analysis	Real-time
Detection of Known Attack	High	Low
Detection of Unknown Attack	Low	High

4. DEEP LEARNING FOR NETWORK INTRUSION DETECTION

Deep learning is a sub-set of ML (Machine Learning), which is a subfield of AI (Artificial Intelligence). It mimics the human brain in processing data and generating patterns for making advanced decisions. It is also known as a deep neural network and deep neural learning and is concerned with ANN (Artificial Neural Networks), which are algorithms inspired by the structure and function of the human brain. Deep neural networks, CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), and deep belief networks are a few deep learning architectures. It has many applications, including NLP (Natural Language Processing), Computer Vision, social network filtering, speech recognition, bioinformatics, and network intrusion detection.

It is preferred over machine learning when the data is unstructured and colossal. It often stacks up deep hierarchies of non-linear features because complex features cannot be learnt from a shallow architecture. Thus, it can cater to a larger cap of challenges with greater ease and efficiency. Neural networks comprise nodes analogous to neurons in three layers: input, hidden, and output layers. The input or visible layer consists of neurons representing an input based on the information that is to be predicted or classified. There can be one or more layers of nodes or neurons between the input and output layers, known as hidden layers. The output layer comprises the nodes that provide the output variables.

It can be supervised, for example, Image classification [16], face recognition [15], etc., or unsupervised, as in Word embedding, image encoding into lower or higher dimensional, etc. Deep Learning is playing a crucial role in network intrusion detection because of its ability to extract complex features from the feature set, proficiency in detecting unseen attacks, and faster detection than the signature or rule-based systems. Below are some of the models that have been implemented that are proven effective to tackle the problem of network intrusion detection systems.

4.1 LuNet: A Deep Neural Network for Network Intrusion Detection

LuNet is the modified version of HAST-IDS [1]. HAST-IDS is a hierarchical network of CNN (Convolutional Neural Network) [17] to unsheathe spatial features and RNN (Recurrent Neural Network) [18], to capture temporal features from network data. LuNet is based on the same concept, but whereas HAST-IDS works on the basic principle of stacking all RNN layers after stacking the pile of CNN layers, LuNet emphasizes on the hierarchy of combined CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) layers. In HAST-IDS, the CNN hierarchy, which is placed before the RNN hierarchy, may lead to loss of temporal information embedded in the raw input data, which in turn makes the RNN inefficient. On the contrary, LuNet synchronizes both CNN and RNN learning into multiple steps to capture both spatial and temporal features competently from the network traffic. Each step is performed by a combined block of CNN and RNN, referred to as the LuNet block.

The learning granularity of this model is measured in terms of total filters used in the CNN/RNN network. CNN generated a feature map, which is further processed by an activation function, ReLu [21] and then pooling down-sample it to trim off trifling data. To enhance the learning, Batch Normalization has been incorporated to address the issue of covariance shift, which might occur due to the dynamical changes in the input value range from one layer to another. And trainable parameters are used to optimize and update the weights of the network in the learning process to yield a better learning outcome.

Long Short Term Memory or LSTM [19] is used for RNN to extricate temporal features in input data. It has been used instead of traditional RNN because in traditional RNN, learning error accumulates in long term dependencies whereas only persistent features are retained in LSTM, and short-lived errors are dropped off. Since granularity changes from coarse-grained to fine-grain from one LuNet block to another, an extra layer has to be added to reshape the output size of one level, which is expected as input for the next level. In case of overfitting, i.e., when the network has learned training data to restrict its ability to recognise deviants in a new sample, LuNet has used a dropout layer with 0.5 default value after the CNN+RNN hierarchy. In the end, an additional CNN layer and a global average pooling layer extract supplementary spatial and temporal features learned from the LuNet blocks.

The model has been tested on two non-redundant datasets NSL-KDD [6] and UNSW-NB15 [7]. Validation Accuracy (AccV), Detection Rate (DR), and False Positive Rate (FPR) are used as LuNet model evaluation metrics. Both binary and multi-class classifications have shown how LuNet outperforms pre-existing ML and DL techniques with a fair margin. It can effectively detect majority of the Normal traffic, Exploits, Generics, DoS, Shellcode, and Reconnaissance attacks and can moderately discover the Analysis attacks with low FPR. However, it lags in detecting the Backdoor and Worm attacks.

In addition to having a high detection capability level compared to the state-of-the-art network intrusion detection techniques, it also minimizes the false positive-alarm rate.

4.2 Pelican: A Deep Residual Network for Network Intrusion Detection

Pelican is a deep neural network based on residual learning. The basic principle is to incorporate the CNN and RNN in the sub residual network to capture the spatial-temporal features in the input effectively. HAST-IDS [1] and LuNet [2] shows that the performance degrades with the increase in the network depth. The residual learning design can mitigate this since it enforces direct output-input mapping through a shorter route to avoid the vanishing/exploding problem of gradient [20] caused by the existing long propagation path. Pelican's basic residual block is based on the LuNet block since the LuNet block can extract the input data's spatial-temporal features.

The Pelican comprises four learning layers, two Batch-Normalization [8] layers, one Convolution layer, and a reshape layer. Batch-Normalization layers to reduce the internal covariance shift and to refine the learning speed. Convolution layer to extract spatial features and initiate feature map to be processed by activation function ReLU then passed through Max pooling[22] layer to generate the feature map which contains the most prominent features of the former feature map. RNN layer, implemented with GRU [23] (Gated Recurrent Unit, a simplified LSTM) to draw out the traffic data's temporal features. An activation function, tan h, and a recurrent function, sigmoid, have been used here. A reshape layer was also there to keep the output size of one level expected as input for the next level in accordance. To address overfitting [24], a dropout layer [25] is incorporated which randomly drops out a few connections from the network.

Two plain and two residual networks [26] of different depths were constructed to analyze residual networks' efficacy.

1. 21-parameter- layer plain network (Plain-21), built of five LuNet blocks, a global average pooling layer, and a dense layer for the final learning output
2. 21-parameter- layer residual network (Residual-21), built of five residual blocks, a global average pooling layer, and a dense layer.
3. 41-parameter- layer plain network (Plain-41), built of ten LuNet blocks, a global average pooling layer, and a dense layer.
4. 41-parameter- layer residual network (Residual-41), made of ten residual blocks, a global average pooling layer, and a dense layer.

The performance metrics used to evaluate the Pelican are Validation Accuracy (AccV), Detection Rate(DR), and False Alarm Rate(FAR).

NSL-KDD and UNSW-NB15 datasets were used for training the model. The four networks' training losses indicate that Plain-21 has fewer losses than Plain-41 signifying that with the increase in layers, the performance degrades. However, for the networks of the same number of layers, residual learning models have significantly lower losses. The deeper the residual network, the smaller the losses, the higher the detection rate, and the lesser the false alarms. It is perceptible from the results that the residual networks surpass the plain networks and that as the network depth increases, the performance enhances. However, the deep network models need extensive data to avoid overfitting, so Pelican's learning performance can be further improved and evaluated on larger datasets.

The application of residual learning on LuNet blocks and the construction of deeper architectures enhanced the detection rates. They reduced the false alarm rates significantly compared to state-of-the-art techniques.

4.3 Dual Net: Locate then Detect Effective Payload with Deep Attention Network

Dual Net makes use of CNN and RNN architecture with the self-attention mechanism. It was called Dual Net because of the division of the model into two stages, namely feature extraction and feature learning stage. The first or the feature extraction stage extracts the spatial and temporal features, and the second or the feature learning stage improves the overall detection efficiency of the model. Dual Net was devised with the goal to create a model for real-time detection that has a high detection rate, can be trained easily and is lightweight for fast execution.

They have used Depth-wise Separable CNN (DSC) for the feature extraction stage for extracting spatial features. Compared to primitive Convolution nets, DSC[27] is faster as it divides the whole convolutional step into pointwise and depth-wise, resulting in fewer multiplications and less trainable parameters. They have also used the Gated recurrent unit (GRU) and DSC for extracting temporal features. GRU is a type of simplified LSTM used for learning long term dependencies. Their feature extraction stage consists of a dense block and a transition block, and the dense block consists of many plain blocks. A plain block is nothing but a combination of DSC, GRU, batch normalization, max pooling, and dropout layer. In their paper, they have used four plain blocks in the dense block. Every plain block receives the concatenation of the output of all the preceding plain blocks and the input data through shortcut connections as its new inputs. A growth rate k was defined to describe no of plain blocks in a dense block. After the dense block, they also had a transition block to reduce the dimensionality of the features generated to build a deeper network. After the feature extraction stage, there is a feature learning stage, which uses self-attention. The idea is to give a score to every attribute or feature of input. The higher the score of the feature is, the more important that feature is. The model then uses these attention scores to focus on the elements that contribute more in generating output.

To achieve the state of their model's art performance, they tested their model on two datasets NSL-KDD and UNSW-NB15. They used Testing Accuracy(AccT), Detection Rate(DR), and False Alarm Rate(FAR) as their model evaluation metrics. They have shown that their model outperforms all existing ML and DL techniques with a fair margin. However, they have also demonstrated that the model performance degrades if the network is too deep, and again, there were optimization difficulties with deeper resnets.

This model can be deployed for real-time detection and stopping the intruders, but the main problem with this model is that it can only classify known attacks; if an unknown attack comes, it will not classify it.

4.4 Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection

Kitsune is a lightweight NIDS (Network-based Intrusion Detection System) that uses an ANN for online data processing. Since learning an extensive feature set requires much time, which is not feasible in an online setting, ANNs are considered computationally expensive. It has an

unsupervised algorithm that does not require labels to perform classification. Kitsune can perform online processing. It has low complexity because an ensemble of autoencoders, each of low complexity, is training and classifying the data.
The experiments were run by the team [4] on a single core Raspberry Pi and Kitsune proved to be negligible on RAM. It detracts from the complexity inherent in deep learning algorithms. The framework breaks down a traffic instance T into a characteristic set of vectors V where every vector in V is processed in a specific autoencoder. Instance T is discarded after processing. Every autoencoder learns only a small subdivision of the features, making it computationally inexpensive compared to a single ANN. Kitsune has two modes- training and execute. The ensemble of autoencoders [28] process each packet, and there is no output in the training mode. In contrast, every packet is processed, and the output notifies the system if the packet is anomalous or not in the execute mode. Mirsky and team [4] generated the data with real IP cameras, IoT devices, and PCs. Scanning, Denial-of-Service (DoS), man-in-the-middle, and botnet malware, etc., attacks were performed. It was compared to a baseline online Network Intrusion Detection System and offline algorithms such as Isolation Forests and Gaussian Mixture Models. It used simple service discovery protocol (SSDP) flood, Operating System (OS) scan, Mirai, Video Injection, and Secure Sockets Layer (SSL) datasets.

Kitsune outperformed the online model, as well as the offline algorithms, which often have more time to learn so could perform better. [4]

4.5 AnomalyDAE: Dual Autoencoder for Anomaly Detection on Attributed Networks

Anomaly detection is performed by Anomaly Dual AutoEncoder or AnomalyDAE on attributed networks to recognize deviation in behaviour or pattern of nodes when compared to reference nodes. It comprises two autoencoders- a structure autoencoder and an attribute autoencoder which are used to learn both node embedding and attribute embedding near densely populated similar data points of compressed data [5]. The attention mechanism is used in the structure encoder to learn the importance between a node and its neighbor nodes to efficiently capture the structure patterns for anomaly detection. The cross-modality interactions between network structure and node attribute are learned during the reconstruction of node attribute by taking the attribute and node embedding as inputs to the attribute decoder. The reconstruction errors of nodes from the structure, as well as the attribute perspectives, are measured to detect the anomalies. However, the effectiveness of the proposed method is demonstrated by the extensive experiments on real-world datasets. [5]

AnomalyDAE can be deployed in real-time for applications such as IDS (Intrusion Detection Systems), System fault diagnosis, and Opinion spam detection. Despite its heuristic success, some processes are performed in an unsupervised scenario because of ground truth anomalies' expensive labeling costs. Though the real-time anomaly detection makes it even harder to sustain against the ground truth anomalies as the labeling, reconstruction of outputs also makes it challenging for the state-of-the-art machines to perform. The model was trained and evaluated on some real-world datasets [36], including Blog Catalog, Flickr, and ACM. Moreover, since new kinds of anomalies may frequently arise over time, it brings further challenges to conventional anomaly detection algorithms as they are often applied in a batch setting and are incapable of interacting with the environment [5].

4.6 Spiking Neural Networks with Single-Spike Temporal-Coded Neurons for Network Intrusion Detection

A new class of neural networks has recently been discovered, known as spiking neural networks [29] (SNN). These SNNs are also known as 3rd generating machine learning techniques. They are different from existing artificial neural networks (ANN) in terms of architecture and how they process information. The motivation behind artificial neural networks was the human brain. The aim was to mimic the thinking process of a human. Compared to the real human brain, ANNs are biologically the inaccurate representation of the human brain incapable of mimicking the human brain mechanism. Whereas, on the other hand, SNN was devised using biologically realistic models of human brain neurons. They operate using spikes, which are nothing but discrete events taking place at a point in time. A spike's occurrence is determined by those differential equations that represent various biological processes.

Despite their robust architecture and real human neuron-like structure, the performance of the SNNs is not that good compared to existing ANNs. It is due to the fact that there is no correct way of training the SNNs. Till now, since spike trains are not differential, so we cannot train an SNN using any of our existing optimization algorithms (Gradient descent, RMS prop, Adam Optimization, etc.). There are ways to train an SNN using Hebbian learning and STDP [31], but their results are not acceptable compared to existing ANNs. Apart from all the difficulties Shibo Zhou and Xiaohua Li [14] have tried to detect Network intrusions using SNNs. In this paper, SNNs were used with a general class of single spike temporal-coded integrate and fire neurons. NSL-KDD [6] [37] and the AWID [38] datasets were experimented upon to examine the input-output expressions on both leaky and non-leaky neurons. It has drawn out the conclusion that mostly adopted SNNs are overly complex and overly nonlinear, leading to slow training and ill convergence, so simplifying every neuron's input-output nonlinearity enhances both training speed and convergence [30]. Lastly, they also provide a training algorithm for training SNNs with non-leaky neurons; by using this algorithm, SNNs can also be trained easily like conventional ANNs.

New SNNs outperformed Logistic Regression [32], SVM, KNN [33], Random Forest, Decision Trees, Ada boost [34], Naive Bayes, Neural Network, CNN-1D, and Reinforcement Learning [35] in both NSL-KDD and AWID dataset

Table 3. Comparative Study of Models

Model	Description	Advantages	Disadvantages
LuNet	● Learning Methodology: Supervised learning technique. ● Architecture: Uses a hierarchy of combined CNN and RNN layers. ● Uses LSTM in RNN layer. ● Dataset: Trained and evaluated on NSL-KDD and UNSW-NB-15	Temporal features are not lost since it uses a combination of CNN along with RNN.	If the model becomes too deep, information may get lost. It can only classify known attacks.
Pelican	● Learning Methodology: Supervised learning technique. ● Architecture:- Consist of plain blocks and residual networks. ● A plain block is a combination of batch normalization, CNN, RNN, and dropout. ● Uses GRU in RNN layer. ● Dataset: Trained and evaluated on NSL-KDD and UNSW-NB-15.	Preserves the information with the help of residual networks.	Need extensive data to avoid overfitting. It is also able to classify known attacks only.
Dual Net	● Learning Methodology: Supervised learning technique. ● Architecture:-Model has two stages-feature extraction, and feature learning. ● Feature extraction:- No: of plain blocks are used and each plain block receives input from all previous plain blocks. ● Plain block is a combination of CNN, RNN, batch normalization, and dropout. ● DSC used in CNN and LSTM used in RNN. ● Feature Learning:- Self-attention mechanism is used. ● Dataset: Trained and evaluated on NSL-KDD and UNSW-NB-1.	More important features are selected while training with the help of self-attention mechanism.	Excessive use of plain blocks may lead to poor outcome. It can also classify known attacks only.
Kitsune	● Learning Methodology: Unsupervised learning technique. ● Architecture:- Uses an ensemble model of autoencoders, and another autoencoder is used as a voting mechanism. ● Datasets:- Tested and evaluated on their own dataset.	It is very lightweight and can be used in a plug and play manner. Can classify unknown attacks also because of its ability to focus on normal traffic rather than malicious traffic.	Its performance is not so good on many of the attacks.
AnomalyDAE	● Learning Methodology: Unsupervised learning technique ● Architecture:- Uses two autoencoders, one for node embedding and one for attribute embedding. ● The self-attention mechanism is used in structure autoencoder. ● Datasets:- Trained and evaluated on Blog Catalog, Flickr, and ACM.	Can classify unknown attacks. More important features can be selected as it also uses a self-attention mechanism.	Very slow for implementation in real-time.
SNN model	● Learning Methodology: Supervised learning technique. ● Architecture:- SNNs with a general class of single spike temporal-coded integrate and fire neurons. ● The complexity of each neuron was decreased for fast training. ● Datasets:- Trained and evaluated on NSL-KDD and AWID.	SNNs are inspired by biologically realistic models of the human brain, so their computation method is better than ANNs.	Difficult to train as these are not trained by traditional optimization algorithms. Results of SNNs are not good as compared to traditional ANNs

5. COMPARATIVE ANALYSIS

The three models Pelican, Dual Net and Lunet are evaluated on the same evaluation metrics, that are Validation accuracy(ACC), Detection Rate(DR) and False Alarm Rate(FAR). ACC is the ability of the model to correctly classify the attacked and non-attacked traffic, DR or True positive rate is the ability of the model to correctly identify the attack traffic and FAR or False positive rate tells us that how many time model misclassified the normal traffic as attack traffic, the detailed definition of the formulas are given in [i], [ii] and [iii]

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad [i]$$

$$DR = \frac{TP}{TP + FN} \quad [ii]$$

$$FAR = \frac{FP}{FP + TN} \quad [iii]$$

Where TP stands for True Positive, TN stands for True Negative, FP stands for False Positive and FN stands for False Negative.

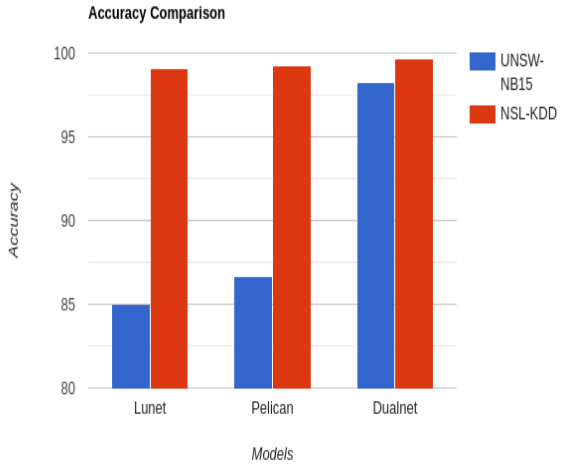

Figure 3. Comparison on the basis of Accuracy

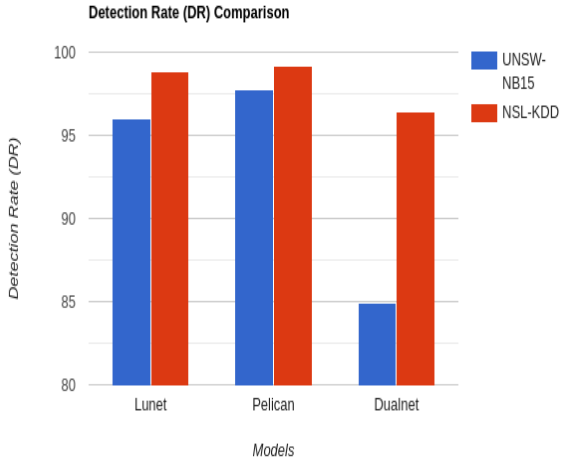

Figure 4. Comparison on the basis of Detection Rate

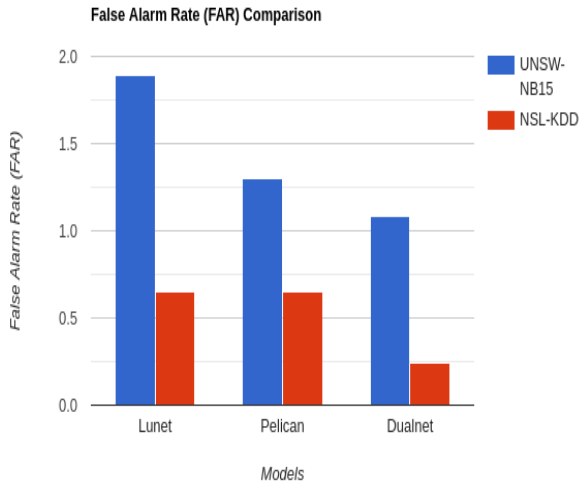

Figure 5. Comparison on the basis of False Alarm Rate

All these models are evaluated on the UNSW-NB15 dataset and NSL-KDD dataset. Figure3 shows the comparison of the all three models on the basis of validation accuracy on both the datasets. In this case, Dual Net outperforms both the models, which means that Dual Net has the best overall accuracy. Figure 4 emphasizes on the comparison of the all three models on the basis of the detection rate, and in this case, Pelican outperforms both the models, which means that Pelican detects the attacks most of the time correctly. Figure 5 shows the comparison of the all three models on the basis of the False alarm rate, in this case, Dual Net has the lowest false alarm rate which means that Dual Net raises less false alarms by categorizing normal traffic as attack traffic as compared to other two models.

6. CONCLUSION

The idea of this paper was to study and review different approaches, supervised and unsupervised techniques that were applied using deep learning to enhance the network intrusion detection systems (NIDS). There are some models that can be used significantly for real-time detection. However, the problem lies within the available datasets. With the ease of data available these days, it is challenging to prepare a dataset to train the model for real-time deployment. New types of hacks are devised every day, making it challenging to keep track of recent cyber-attacks. But unsupervised learning methods can overcome this problem because of their ability to focus on standard data rather than odd data. It is not preferable to depend upon machine learning or deep learning methods alone because their ability to detect cyber-attacks is not ideal.

If a deep learning system is trained for real-time deployment, even with 99% accuracy, there is still a 1% chance that the network could be compromised. Then it wouldn't be ideal to use it for security measures. It may lead to the loss of precious data that is not at all acceptable.

7. REFERENCES

[1] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, “Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection,” IEEE Access, vol. 6, pp. 1792–1806, 2017.

[2] Wu, Peilun & Guo, Hui. “LuNet: A Deep Neural

Network for Network Intrusion Detection. 617-624. 10.1109/SSCI44817.2019.9003126." ,The University of New South Wales, 2019

[3] Wu, Peilun & Guo, Hui. "Pelican: A Deep Residual Network for Network Intrusion Detection.",School of Computer Science and Engineering, University of New South Wales, Sydney, 2020

[4] Mirsky, Yisroel & Doitshman, Tomer & Elovici, Yuval & Shabtai, Asaf."Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. 10.14722/ndss.2018.23211.", Ben-Gurion University of the Negev, 2018

[5] Fan, Haoyi & Zhang, Fengbin & Li, Zuoyong. "Anomalydae: Dual Autoencoder for Anomaly Detection on Attributed Networks. 5685-5689. 10.1109/ICASSP40776.2020.9053387."School of Computer Science and Technology, 2020

[6] KDD Cup 99, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html ."

[7] UNSW-NB15dataset, http://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015

[9] Evans, D.: The Internet of Things How the Next Evolution of the Internet Is Changing Everything (2011)

[10] I. Rish,"An empirical study of the naive Bayes classifier", T.J. Watson Research Center

[11] S. Rasoul Safavian and David Landgrebe, A Survey of Decision tree Classifier Methodology

[12] S.V.N. Vishwanathan, M. Narasimha Murty, "SSVM : A Simple SVM Algorithm",{vishy, mnm}@csa.iisc.ernet.in Dept. of Comp. Sci. and Automation

[13] Shiyi Yang, Peilun Wu and Hui Guo,"DualNet: Locate Then Detect Effective Payload with Deep Attention Network", School of Computer Science and Engineering, University of New South Wales, Sydney(2020)

[14] Shibo Zhou and Xiaohua Li, "Spiking Neural Networks with Single-Spike Temporal-Coded Neurons for Network Intrusion Detection", {szhou19, xli}@binghamton.edu (2020)

[15] Faizan Ahmad, Aaima Najam and Zeeshan Ahmed, "Image-based Face Detection and Recognition: "State of the Art"

[16] Xin, M., Wang, Y. Research on image classification model based on deep convolution neural network. *J Image Video Proc.* 2019, 40 (2019). https://doi.org/10.1186/s13640-019-0417-8

[17] Saad ALBAWI, Tareq Abed MOHAMMED, "Understanding of a Convolutional Neural Network", ICET2017, Antalya, Turkey

[18] Alex Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network", arXiv:1808.03314v8 [cs.LG] 21 Dec 2020

[19] Klaus Greff, Rupesh K. Srivastava, Jan Koutn´ık, Bas R. Steunebrink, Jurgen Schmidhuber, "LSTM: A Search Space Odyssey", arXiv:1503.04069v2 [cs.NE] 4 Oct 2017

[20] Hochreiter, Sepp. (1998). "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems." 6. 107-116. 10.1142/S0218488598000094.

[21] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall,"Activation Functions: Comparison of Trends in Practice and Research for Deep Learning", arXiv:1811.03378v1 [cs.LG] 8 Nov 2018

[22] Nagi, Jawad & Ducatelle, Frederick & Di Caro, Gianni & Ciresan, Dan & Meier, Ueli & Giusti, Alessandro & Nagi, Farrukh & Schmidhuber, Jürgen & Gambardella, Luca Maria. (2011). "Max-pooling convolutional neural networks for vision-based hand gesture recognition". 2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011. 342-347. 10.1109/ICSIPA.2011.6144164.

[23] Rahul Dey and Fathi M. Salem, "Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks"

[24] Xue Ying 2019 J. Phys,"An Overview of Overfitting and its Solutions", Conf. Ser. 1168 022022

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov,"Dropout: A Simple Way to Prevent Neural Networks from Overfitting"

[26] Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, "Deep Residual Learning for Image Recognition", arXiv:1512.03385v1 [cs.CV] 10 Dec 2015

[27] Franc¸ois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", arXiv:1610.02357v3 [cs.CV] 4 Apr 2017

[28] Dor Bank, Noam Koenigstein, Raja Giryes, "Autoencoders", arXiv:2003.05991v1 [cs.LG] 12 Mar 2020

[29] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothee Masquelier, and Anthony Maida, "Deep Learning in Spiking Neural Networks", arXiv:1804.08150v4 [cs.NE] 20 Jan 2019

[30] Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[31] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," Neural Networks, vol. 99, pp. 56–67, 2018.

[32] Peng, Joanne & Lee, Kuk & Ingersoll, Gary. (2002). An Introduction to Logistic Regression Analysis and Reporting. Journal of Educational Research - J EDUC RES. 96. 3-14. 10.1080/00220670209598786.

[33] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification.

[34] Chengsheng, Tu & Huacheng, Liu & Bing, Xu. (2017). AdaBoost typical Algorithm and its application research. MATEC Web of Conferences. 139. 00222. 10.1051/matecconf/201713900222.

[35] Leslie Pack Kaelbling, Michael L. Littman, "Reinforcement Learning: A Survey"

[36] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu, "Deep anomaly detection on attributed networks," in Proceedings of the 2019 SIAM International Conference on Data Mining. SIAM, 2019, pp. 594–602.

[37] "Nsl-kdd data set for network-based intrusion detection systems." http://nsl.cs.unb.ca/KDD/NSL-KDD.html, March 2009.

[38] Kolias, Costantinos & Kambourakis, Georgios. (2015). Aegean Wi-Fi Intrusion Dataset (AWID), http://icsdweb.aegean.gr/awid