International Journal of Information Movement *Vol. 9 Issue III* *(July 2024)*
Website: www.ijim.in *ISSN: 2456-0553 (online)* **Pages 48-57**

TECHNOLOGICAL AND STATUS REVIEW OF DEEP LEARNING ALGORITHMS AND THEIR APPLICATIONS IN CYBER SECURITY

Seema Jain,
Ph.D. Research Scholar
Department of Computer Science and Engineering, School of Engineering and Technology,
Raffles University, Neemrana

Dr. Shiv Kant
Associate Professor
Department of Computer Science and Engineering, School of Engineering and Technology,
Raffles University, Neemrana
Email-Id: dr.shivkant@rafflesuniversity.edu.in
and
Dr. Aman Ahmad Ansari
Associate Professor
Department of Computer Science and Engineering, School of Engineering and Technology,
Raffles University, Neemrana

Abstract: There have been an increasing number of cyber-attacks due to the exponential growth of the Internet as a result of advances in technology. Every industry and field of study is beginning to integrate Deep Learning applications. The field of cyber-security looks to benefit significantly from Deep Learning (DL), one of the most effective machine learning techniques available today. This study provides a description of deep learning (DL) techniques for cyber-security applications, discusses the security concerns with DL networks, and shows how these models could be used maliciously. A concise overview of the architecture covered in this work is given at the outset. Various models, such as CNNs, RNNs, and GANs, have security-related indications.

Keywords- Deep Learning, Cyber security, attacks, Application, DL techniques

1.0 Introduction

Security measures are becoming more important as the number and variety of computing devices in use across the globe continues to grow. The use of artificial intelligence has skyrocketed in tandem with the proliferation of related technologies. There are a number of risks associated with the use of artificial intelligence (AI) in cyberspace, including the fact that many practical applications rely on AI, which in turn creates security concerns, and the fact that AI has already formed the basis for a number of security applications. This paper delves into the role of deep learning in cybersecurity, highlighting its usefulness while also highlighting its potential for malicious usage, which can affect both people and organizations. It also discusses the security challenges that arise from deep learning applications. The idea of reciprocal trust has made internet a dangerous & untrustworthy place in the modern day. Actually, the premise that users in this type of network are mutually trustworthy was fundamental to the construction of the Internet's architecture. This model does not require security. Unfortunately, our assets, privacy, and lives are seriously endangered by a wide variety of threats, including Trojan horses, worms, viruses, spoofing, distributed denial of service, and computer and network monitoring. The security concerns of real-world applications & uses and exploits of Deep Learning in the cybersecurity sector are the exclusive focuses of this study.

2.0 Deep Learning Architectures

According to J. Schmidhuber (2015), deep learning architectures like these are based on neural networks, which are an approach to machine learning that mimics the way the brain's neurons communicate with one another. Each of these networks has an input layer, an output layer, and a hidden layer in between. Neurons, the building blocks of computing, receive data as input, process it using mathematical operations called activation functions, and then either represent the result as the final output (in the output layer) or send it on to the next layer.

2.1 Deep Neural Networks: According to Zell (1994), a Deep Neural Network (DNN) is a type of feed-forward

neural network that has multiple hidden layers and node connections that do not form a cycle. From the input layer to the output layer, via one or more hidden layers, these networks adhere to a linear data flow. As long as there are enough layers, these networks can map complicated functions. Text classification, anomaly detection, & financial sales prediction are just a few examples of the many classification tasks that see widespread use of DNNs.

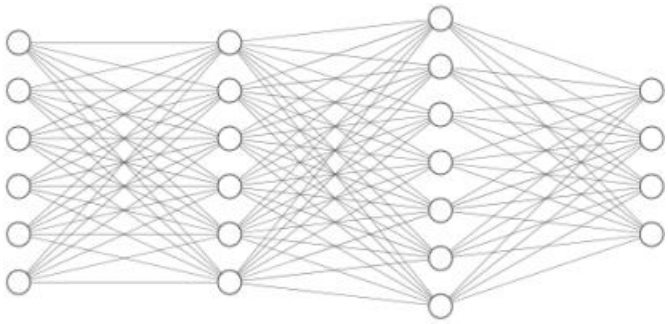

Fig. 1. Deep Neural Network

2.2 Convolutional Neural Networks: Among the many types of deep learning systems, Convolutional Neural Networks (CNNs) stand out for their ability to aggregate data processing components called kernels. In contrast to neural networks, which only examine individual data points, these kernels allow them to extract features from the data throughout a region. The majority of CNN designs, including ResNet from Microsoft and InceptionV3 from Google, have been developed for use in computer vision applications, such as object categorization using the ImageNet Dataset (Deng, J., Dong 2009; He, K., Zhang 2016; Szegedy 2016).

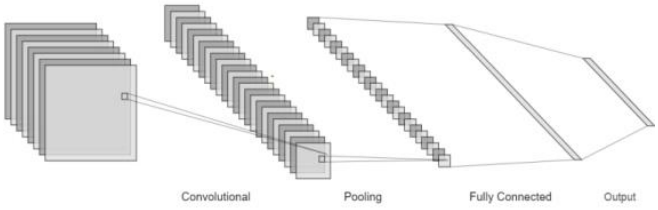

Fig. 2. Convolutional Neural Network

The convolutional, pooling, and fully linked layers make up a standard CNN architecture (Fig. 2). The convolutional layer is responsible for running the kernel function. Each kernel stands for a different data characteristic because the model learns its weights during training. In order to calculate an average & decrease the input size, pooling layers are included between convolution layers. The convolutional layer's output is smoothed down by the fully connected layer, which then uses the features to conduct classification. It typically uses a DNN but can also incorporate recurrent neural networks (RNNs) to create hybrid deep learning models.

2.3 Recurrent Neural Networks: Deep learning models known as Recurrent Neural Networks (RNNs) incorporate the output of an earlier input into the current input, in contrast to traditional DNNs. Therefore, tasks involving sequential analysis, such speech recognition, handwriting recognition, or semantic analysis, make use of these networks' ability to evaluate sequential input. The vanishing gradient problem is a difficulty with RNNs. It happens when the change in weights stops or becomes inconsequential during the backpropagation/learning phase because the total of small partial derivatives is so small. Long short-term memory (LSTM) networks are employed to address this. A "forget gate" component, absent in RNNs, allows these networks to collectively determine the significance of prior knowledge by assigning a single value instead of the product.

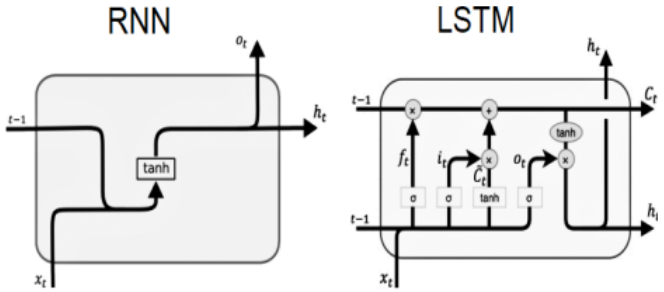

Fig. 3. RNN unit vs LSTM unit

2.4 Generative Adversarial Networks: Goodfellow et al. (2014) created generative neural networks, or GANs. Two models, a generator and a discriminator, are trained simultaneously to make them. The discriminator examines the output of the generator and finds out how closely it matches the features of the actual output; the generator then tries to produce the intended output from the input. Backpropagation & log loss functions are used to train both models. With the development of numerous versions, such as CycleGAN (Zhu, J. Y., Park 2017), contextual GAN (Lu, Y., Wu, S 2018), & VAE-GAN (Larsen 2016), GANs have gained immense popularity since their beginnings. picture production, picture style transfer, and 3D object synthesis are some of their more noticeable uses in computer vision.

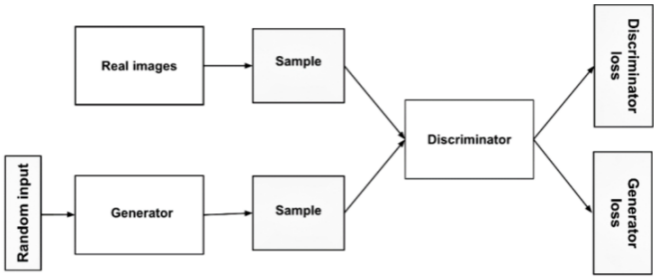

Fig. 4. GAN architecture

2.5 Autoencoders: Using a series of hidden layers, an autoencoder is a kind of neural network that attempts to reconstruct the input at the output layer. Autoencoders work by learning an encoding, or representation, for a given dataset. It achieves this by extracting a network's hidden layer & uses it to represent data. The autoencoder successfully decreases the data's dimensionality when this layer's size is smaller than the input. Encoding data and compressing it are two common uses for autoencoders. They are also known as denoising autoencoders when they're utilized to remove noise from input. In addition to its use in feature extraction and picture classification, autoencoders can also be employed in image synthesis.

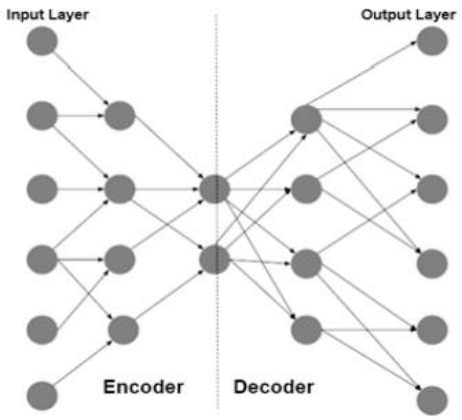

Fig. 5. Autoencoder architecture

3.0 Dl Techniques For Cybersecurity

Deep (Feedforward) Neural Networks: An artificial neural network's simplest form is the feedforward neural

network. Due to their ability to estimate complex functions, given sufficient layers, DNNs are highly effective for classification and regression applications, despite being the simplest neural network architecture. (Manuel 1992) Detecting malware, threats, & technological abuses is an area where DNNs have demonstrated strong performance in cybersecurity. Because of their capacity to learn new patterns and provide probability rather than a definitive answer, Cannady (1998) suggested utilizing neural networks for misuse detection. As an indication of their practical utility, he trained a model neural network that achieved an accuracy level of more than 90% for the chosen test classes.

Wu (2009) classified spam emails using a hybrid approach that included rule-based and neural network methods. Instead of looking at keywords in the content, their approach focused on the more consistent factor in spam e-mails, which are intrinsic spamming habits. With a precision of 99.60%, the model surpassed numerous other modern networks, including NaiveBayes, ADTrees, and Bayesian Networks. A new method for identifying Zombie PCs utilizing neural networks was proposed by Salvador et al. (2009). They planned to use a neural network trained on simulated data to examine the network traffic. With a minimum accuracy of 87.34%, the model produced satisfactory results for a range of traffic scenarios. Using a unique architecture that included both DNNs & RNNs, Tuor et al. (2017) were able to detect insider threats. Results from experiments shown that their models outperformed state-of-the-art anomaly detection methods, such as SVMs & PCA. Almost all harmful occurrences were classified as above-average by the DNN, with the majority falling above the 95th percentile.

Additionally, intrusion detection systems have made use of DNNs. Applications of DNNs to intrusion detection systems were investigated by Bitter et al. (2010). For intrusion detection systems (IDS), denial-of-service (DoS) assaults, and spam detection, DNNs were compared against state-of-the-art. The average accuracy of DNNs utilized for intrusion detection systems was 83.14%. They came to the conclusion that these networks worked effectively and could be used to construct strong IDS even under less-than-ideal conditions. In order to detect cache side-channel attacks, Depoix (2018) employed DNNs. A computer system's inner workings can be inferred & exploited using side-channel attacks, which monitor changes in the system's environment. The authors used neural networks to monitor the CPU's cache activity and identify malicious or benign cache accesses based on HPC features. The researchers attained a precision level of 99.23%. Lastly, some security applications were addressed by Vinayakumar et al. (2018) through the use of DNNs. In order to classify malware, they used APKs from the Opera Mobile store; to detect incidents, they utilized UniteCloud UTM; and to detect fraud, they utilized unified, anonymized data obtained by HCRUD. At least 94% of the time, the DNN models' classification accuracy was higher than XGBoost's in every use case.

3.1 CNNs : The separate kernel layers that conduct the convolutions make convolutional networks ideal for image processing jobs. Nevertheless, by transforming input information into images and putting them into the convolutional model, these systems can be utilized for classification tasks in cybersecurity. In order to identify domain names that are the result of domain generation algorithms (DGAs), Yu et al. (2017) put forward a new method. In order to establish connections to their command & control servers, botnets use DGAs. For real-time DGA identification, the authors trained two deep networks—a CNN and an LSTM—with an embedding layer each. The networks were then tested on real huge traffic. AUCs of 0.9918 and 0.9896 were attained by CNN & LSTM, respectively. In terms of false positive rate (FPR), the authors found that the CNN model fared the best at 0.01%. Several models, including ResNet, AlexNet, and Inception, that were pre-trained on ImageNet were used to detect DGAs by Zeng et al. (2017). A decision tree classifier was fed the output of the retrained models after they were modified to accept string inputs. They used the Inception V4 model & got the greatest results with a 99.56% accuracy, a 99.86% true positive rate, and a 1.128% false positive rate. By combining natural language processing (NLP), convolutional neural networks (CNN), and long short-term memory (LSTM) classifiers, Hendler et al. (2018) were able to recognize malicious Powershell commands, thus addressing the problem of Powershell assaults. They got TPRs of 0.92, 0.89, and 0.72 for FPRs of 10-2, 10-3, and 10-4, respectively, by forming an ensemble, which outperformed the non-ensemble techniques. An EDCNN architecture was suggested by Shibahara et al. (2017) as a means of detecting drive-by download attacks. When a victim visits a compromised website or downloads infected software, these assaults insert harmful malware onto their system. Input URL sequences are processed by the event denoising convolutional neural network (EDCNN), which then returns a malicious or benign verdict. Using the least amount of computing time, the EDCNN could identify harmful redirections, categorize sequences with a poor FPR, and more. Using unprocessed data from network traffic and feeding it into a convolutional neural network (CNN), Wang et al. (2017) developed an intrusion detection method. First, we used a 20-class and a 10-class classifier, respectively,

to achieve the classification; these classifiers could only produce two possible outcomes: harmful or non-malicious. The second step was to determine if the result was harmful or not by using binary classification. For the datasets in question, the binary classifier attained a perfect score of 100%, whereas they reached 99.17% and 99.41% accuracy with 20- and 10-class classifications, respectively.

3.2 RNNs : When it comes to sequential data processing, RNNs, especially LSTMs, have exploded in popularity. Since LSTMs can identify continuous patterns like aberrant behavior or harmful attacks, they find use in intrusion detection & anomaly detection in cybersecurity. For malware detection, Kolosnjaji et al. (2016) utilized a CNN/RNN hybrid. They trained the DL model using the call sequences transformed to onehot encoding. An LSTM & CNN with the SoftMax output function made up the model. It outperformed the Support Vector Machines & Hidden Markov Models in terms of accuracy (89.4%), precision (80%), and recall (80%). For the purpose of detecting botnets, Torres et al. (2016) suggested an LSTM consisting of 128 nodes. The model was trained using stratified 10-fold cross validation, & features were one-hot encoded. Oversampling, under sampling, and no sampling at all were the three methods used to choose the data. With a 96% success rate and a 0.0111 average false alarm rate, the oversampling strategy outperformed the others. In their 2018 study, McDermott et al. trained a bidirectional LSTM to identify botnets in Internet of Things devices. In order to generate a model representation from the string data input, the LSTM model included a word-embedding layer. The scientists replicated four of Mirai's assaults—a UDP flood, an ACK flood, a DNS attack, & Synchronize attack—in their own dataset and put the model through its paces. Depending on the attack type, the bidirectional LSTM attained accuracies ranging from 91.95% to a high of 99.999%. The effectiveness of RNNs as intrusion detection systems has been the subject of multiple studies. On the KDD-1999 datasets, Staudemeyer (2015) and Kim and Kim (2015) employed LSTMs & RNNs, respectively. On the other hand, Kim et al. (2016) produced extra data and utilized an ensemble of LSTM classifiers on the KDD-1999. Kim & Kim [2015] achieved the highest results, with a detection rate of 100% and a false alarm rate of 2.3%. The authors Loukas et al. (2017) employed LSTMs to identify different forms of attacks, such as DDoS, malware, & command injection. Their 86.9% accuracy rate was higher than that of the competing algorithms. In tests including unknown data, the LSTM also fared better than competing models. A technique to identify API requests as ransomware was suggested by Maniath et al. (2017). They used integer encoding to perform dynamic analysis on API calls from executables, which were then input into an LSTM network and either ransomware or benign classes were assigned to them. They achieved an impressive 96.67% accuracy using a dataset consisting of 157 distinct ransomware samples of varying lengths. Several publications have suggested RNNs as an authentication mechanism for IoT (Internet of Things) systems. An innovative method for watermarking signals from Internet of Things devices was put out by Ferdowsi et al. (2018). It employed a long short-term memory (LSTM) model that makes advantage of stochastic properties, such as the skewness and spectral flatness, of the signals. Using this watermarked signal, they were able to authenticate signals securely and identify attacks in the system. Fingerpass is the term given to the real-time authentication and security system proposed by Kong et al. (2019) for use in Smart homes. It checks the user's identity by using WiFi signal channel state information to deduce the distinct behavioral traits of finger motions. At the login phase, the LSTM network was employed for user authentication. It accomplished a respectable latency of 800-1200ms, an average FPR of 3.8%, and an accuracy of 92.6% on average.

3.3 GANs: GANs can leverage their genera-tor-discriminator architecture to create images, patterns, or any type of data. As we'll see in the next section, this data-generation capability has been found to result in more harmful than helpful use-cases. Nevertheless, as mentioned below, GANs have had some beneficial uses in security applications: In order to identify anomalies, Zenati et al. (2018) [40] created a GAN model. They tweaked the GAN model such that, in addition to a generator and discriminator, it could use an encoder to generate a latent representation. The discriminator made use of both the learnt generator output and this latent representation throughout training. After that, it was tested on data that could be seen (MNIST) and data that could not be seen (KDD99). With 92% accuracy, 95.78% recall, and an F1 score of 93.72%, their model was competitive with state-of-the-art.

In their 2019 study, Chen et al. employed an intrusion detection model based on GANs with numerous intermediate layers. They used the KDD-99 dataset to evaluate their architecture, which was based on the bidirectional GAN (Bi-GAN). With an F1 score of 93.98%, a recall of 94.73%, & precision of 93.24%, their model surpassed state-of-the-art models. Compared to BiGAN, it sped up testing by a factor of 1.357 and cut training time by 12% from the state-of-the-art, further decreasing the overhead. For the goal of image steganography, Volkhonskiy et al. (2017) suggested GANs. For the purpose of creating picture covers, they employed Deep Convolutional Generative Adversarial Networks (DCGANs). With a reconstruction quality of at

least 98.65% and a DCGAN that can successfully generate cover images, they demonstrated that their network is also suitable for picture encryption. A new approach to secure steganography was suggested by Shi et al. (2017), which involves the use of GANs. One generator & two discriminators make up their suggested model, SSGAN. Covers for steganography are generated by the generator G, images are evaluated for quality by the discriminator D, and the images are steganalysis to determine their suitability by the discriminator S. According to Liu and Luo (2015), the CelebA face dataset was used to train the models. According to Volkhonskiy (2017), they demonstrated a faster convergence time compared to SGAN. Steganalysis revealed that their model had produced a more secure cover since it increased the error rate.

Using GANs, Lee et al. (2017) trained a network that can withstand malicious inputs. They teach a generator to produce adversarial instances & train a classifier to accurately categorize both the original photos and the adversarial examples. While testing on the CIFAR 10 or CIFAR 100 datasets, the authors found that their classifier outperformed the others using dropout, adversarial training, and random perturbation. Although their model outperformed the fast gradient technique in terms of generalization, the authors acknowledged that it was slower and could benefit from additional research to enhance its performance. An concern with generative model picture generation—information leakage—and a way to prevent it are presented by Hayes et al. (2017). Due to data leakage, an attacker may be able to infer details about the training dataset. The authors provide a GAN model for detecting generative network overfitting, and then they compare the produced synthetic images to the actual ones to find out what the network was trained on. On the CIFAR-10 dataset, they succeeded in achieving white-box accuracy of 100%, black-box accuracy of 58% with minimal knowledge, or black-box accuracy of 37% without knowledge. Using the authors' approach, these tests demonstrate that generative network makers can detect overfitting in their networks. To help botnet detection models work better, Yin et al. (2018) suggest Bot-GAN, an architecture that generates false samples. The four distinct kinds of bot-net traffic included in the ICSX botnet dataset were used to train the Bot-GAN. It draws from a pool of 122 features that fall into one of those four categories. As demonstrated by the experimental data, the model successfully enhanced the performance of the initial detector. They saw a decrease in FPR from 19.19% to 15.59% when they used 500 mixed samples.

3.4 Autoencoders: Autoencoders have demonstrated promising performance in categorization and intrusion detection systems. They can be utilized in hybrid models alongside other DL architectures like RNNs & CNNs because of their data reduction capabilities. Malware classification was carried out by Wang and Yiu (2017) utilizing autoencoders based on RNNs. Prior to feeding API call sequences into an RNN for malware classification, they employed autoencoders to lower the dimensionality of the data. On a publicly available malware dataset, they attained a 99.1 percent success rate. They also generated FAPs with a 98.3% success rate using the model for representation learning. In their groundbreaking Deep Packet architecture, Lotfollahi et al. (2017) classified traffic and identified applications by combining CNN with stacked autoencoder (SAE). For application identification, the writers attained an F1 score of 95%, and for traffic characterization, they attained an F1 score of 92%. The authors came to the conclusion that their model can automate feature extraction and traffic categorization. They also want to apply it to classify Tor traffic after additional research.
A stacked autoencoder was trained to identify impersonation attacks by Aminanto and Kim (2017). In these types of attacks, a malicious device masquerading as an authorized access point obtains network access and then executes malicious code. Specifically, they looked at two of the four possible classes—harmless and attack—in the AegeanWiFi Intrusion Dataset's "CLS" dataset (Kolias, C., Kambouraki 2015). In order to extract features from the dataset, a stacked autoencoder with two hidden layers was employed. In order to distinguish between the two types of entities, "impersonating" and "real," a k-means clustering algorithm was given these characteristics. With a recall of 92.18% and a precision of 86.15%, they were able to get an accuracy of 94.81%. An approach to anomaly identification utilizing variational autoencoders (VAE) on different datasets was suggested by Sun et al. (2018). For network anomaly detection, they utilize the KDD-CUP'99 dataset; for image anomaly screening, they employ the MNIST dataset; and for anomaly identification in video surveillance, they use the UCSD Pedestrians dataset. Prior to using the network to rebuild output for test data, they trained it on the dataset to learn how to transfer the hidden characteristics onto a latent variable. This allowed them to detect input abnormalities. To identify data as unusual, we looked at how much it differed from the test input & reconstructed output. With an AUC of 95.1% and an F1 score of 81.2%, their model outperformed or was on par with the top detectors on the KDD dataset.

In their study, Ma et al. (2018) utilized a unique method known as SCDNN to detect intrusions on sensor networks. Spectral clustering or deep neural networks were utilized by the SCDNN classifier to categorize

aspects of intrusion data into five groups: normal, four dangerous, and unidentified. By employing spectral clustering (SC), SCDNN is able to train k deep neural networks (DNNs) to divide the data into k subsets or clusters. The hidden layers of a DNN are built up of many autoencoders. After experimenting with various values for k, the authors found that 2 produced the most stable and optimal model results. Using a three-layer autoencoder, Diro and Chilamkurti detected intrusions on devices and networks connected to the internet of things. By using a master node to update the settings assigned to each worker fog node, they hoped to detect DoS (denial of service) and other forms of assaults on fog-to-things/edge computing systems. An unsupervised stacked autoencoder architecture utilized as the training model. In comparison to shallow learning methods, which only managed a 99.2% accuracy on the KDDCUP99 dataset, their performance was exceptional. The RATAFIA ransomware detection system was created by Alam et al. (2019) and is based on LSTM autoencoders that were trained on typical program behaviors. They combed through the HPC information to determine the executable's impact on the hardware. While other ransomwares were identified in less than 2 seconds, their technology discovered the WannaCry ransomware in 5.313 seconds, according to their decision-making process regarding detection time windows. Plus, RATAFIA doesn't require any kernel tweaks to run on any contemporary CPU that has HPC support. Shi et al. (2017) followed Kong, H., and Lu (2019) in utilizing an autoencoder to detect human behavioural & physiological aspects based on their daily activity patterns using WiFi signals supplied by IoT devices. The autoencoder generates a distinct identifier ("fingerprint") for every user by utilizing this channel state information (CSI). Every one of the autoencoder's three layers served a specific purpose: the first for activity separation, the second for action classification, and the third for fingerprint-based user identification. An SVM layer was also a part of the design for spoofing detection. A 91% authentication accuracy was attained by the model.

4.0 Deep Learning In Cybersecurity

Deep learning has become increasingly important in the field of cybersecurity due to its ability to detect and respond to complex threats in large-scale, dynamic environments. Here are several ways in which deep learning is applied in cybersecurity:

- **Malware Detection**: Deep learning models can analyze file contents and behavior to detect previously unseen malware variants. They can learn patterns indicative of malicious code or behavior, even when traditional signature-based methods fail.
- **Anomaly Detection**: Deep learning techniques can be used to identify abnormal patterns in network traffic or user behavior, which may indicate intrusions or suspicious activity. This is particularly useful in detecting zero-day attacks or insider threats.
- **Intrusion Detection and Prevention Systems (IDPS)**: Deep learning models can enhance IDPS by continuously learning from network traffic data to identify and block malicious activities in real-time. They can adapt to evolving threats and improve detection accuracy over time.
- **Phishing Detection**: Deep learning algorithms can analyze email content, URLs, and user behavior to detect phishing attempts. By learning from large datasets of known phishing emails and legitimate correspondence, these models can effectively identify fraudulent messages.
- **Threat Intelligence Analysis**: Deep learning models can analyze vast amounts of textual data from various sources, such as social media, forums, and dark web marketplaces, to identify emerging threats, vulnerabilities, and attack trends.
- **Behavioral Biometrics**: Deep learning can be applied to analyze user behavior patterns, such as keystrokes, mouse movements, and navigation patterns, to authenticate users and detect anomalies that may indicate unauthorized access or account takeover attempts.
- **Vulnerability Assessment**: Deep learning techniques can assist in identifying potential vulnerabilities in software or systems by analyzing code, configurations, and historical data to predict areas susceptible to exploitation.
- **Security Analytics**: Deep learning models can analyze log data, system events, and other security-related information to detect unusual patterns or anomalies that may indicate security breaches or unauthorized activities.
- **Network Traffic Analysis**: Deep learning algorithms can analyze network traffic patterns to detect anomalies, intrusions, or suspicious activities, enabling security teams to respond promptly to potential threats.
- **Adversarial Attacks Defense**: Deep learning techniques can be used to develop robust models resilient to adversarial attacks, where attackers attempt to evade detection or mislead the system by subtly modifying inputs.

5.0 Conclusion

Deep learning is a widely used algorithm in various domains of cyber security. cybersecurity and its many uses for deep learning algorithms. On top of that, we demonstrated the security risks associated with deep learning by demonstrating how their applications are vulnerable to attack & malicious use. The purpose of this study was to show how deep learning may improve upon or even surpass state-of-the-art cybersecurity solutions, and what role it plays in this field. We recommend that academics & businesses think about how to integrate deep learning into existing security systems. Because of their vulnerability to adversarial & poisoning assaults, neural networks require further investigation into how to strengthen their security. The impact of emerging technologies on deep learning, such as 5G, the internet of things, blockchain, quantum computing, & edge computing, can be the subject of future research in this field. New technology always has its own set of security concerns. Future developments in deep learning architectures, both simple and complicated, will likewise necessitate evaluation.

References

i. Abeshu, A., & Chilamkurti, N. (2018). Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Communications Magazine, 56(2), 169- 175.
ii. Alam, M., Bhattacharya, S., Dutta, S., Sinha, S., Mukhopadhyay, D., & Chattopadhyay, A. (2019, May). RATAFIA: Ransomware Analysis using Time And Frequency Informed Autoencoders. In 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) (pp. 218-227).
iii. Aminanto, M. E., & Kim, K. (2017, August). Improving detection of Wi-Fi impersona-tion by fully unsupervised deep learning. In International Workshop on Information Se-curity Applications (pp. 212-223). Springer, Cham.
iv. Bitter, C., Elizondo, D. A., & Watson, T. (2010, July). Application of artificial neural networks and related techniques to intrusion detection. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
v. Cannady, J. (1998, October). Artificial neural networks for misuse detection. In National information systems security conference (Vol. 26).
vi. Chen, H., & Jiang, L. (2019). GAN-based method for cyber-intrusion detection. arXiv preprint arXiv:1904.02426
vii. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & FeiFei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.
viii. Depoix, J., & Altmeyer, P. (2018). Detecting Spectre Attacks by identifying Cache Side-Channel Attacks using Machine Learning. Advanced Microkernel Operating Systems, 75.
ix. Feng, Z., Shuo, C., & Xiaochuan, W. (2017, December). Classification for DGA-based malicious domain names with deep learning architectures. In 2017 Second International Conference on Applied Mathematics and information technology (p. 5).
x. Ferdowsi, A., & Saad, W. (2018, May). Deep learningbased dynamic watermarking for secure signal authentication in the Internet of Things. In 2018 IEEE International Confer-ence on Communications (ICC) (pp. 1-6). IEEE.
xi. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information pro-cessing systems (pp. 2672-2680).
xii. Hayes, J., Melis, L., Danezis, G., & De Cristofaro, E. (2017). LOGAN: evaluating priva-cy leakage of generative models using generative adversarial networks. arXiv preprint arXiv:1705.07663.
xiii. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recogni-tion. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
xiv. Hendler, D., Kels, S., & Rubin, A. (2018, May). Detecting malicious PowerShell com-mands using deep neural networks. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security (pp. 187-197). ACM.
xv. J. Schmidhuber, "Deep Learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015.
xvi. Kim, G., Yi, H., Lee, J., Paek, Y., & Yoon, S. (2016). LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection sys-tems. arXiv preprint arXiv:1611.01726.

xvii. Kim, J., & Kim, H. (2015, August). Applying recurrent neural network to intrusion detec-tion with hessian free optimization. In International Workshop on Information Security Applications (pp. 357-369). Springer, Cham.
xviii. Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2015). Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communica-tions Surveys & Tutorials, 18(1), 184-208.
xix. Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016, December). Deep learning for classification of malware system call sequences. In Australasian Joint Conference on Artificial Intelligence (pp. 137-149). Springer, Cham
xx. Kong, H., Lu, L., Yu, J., Chen, Y., Kong, L., & Li, M. (2019, July). FingerPass: Finger gesture-based continuous user authentication for smart homes using commodity WiFi. In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Net-working and Computing (pp. 201-210). ACM.
xxi. Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016, June). Autoen-coding beyond pixels using a learned similarity metric. In International Conference on Machine Learning (pp. 1558-1566).
xxii. Lee, H., Han, S., & Lee, J. (2017). Generative adversarial trainer: Defense to adversarial perturbations with gan. arXiv preprint arXiv:1705.03387.
xxiii. Lotfollahi, M., Siavoshani, M. J., Zade, R. S. H., & Saberian, M. (2017). Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Computing, 1-14.
xxiv. Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., & Gan, D. (2017). Cloud-based cyber-physical intrusion detection for vehicles using deep learning. Ieee Access, 6, 3491-3508.
xxv. Lu, Y., Wu, S., Tai, Y. W., & Tang, C. K. (2018). Image generation from sketch con-straint using contextual GAN. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 205-220).
xxvi. Ma, T., Wang, F., Cheng, J., Yu, Y., & Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. Sen-sors, 16(10), 1701.
xxvii. Maniath, S., Ashok, A., Poornachandran, P., Sujadevi, V. G., Sankar, A. P., & Jan, S. (2017, October). Deep learning LSTM based ransomware detection. In 2017 Recent De-velopments in Control, Automation & Power Engineering (RDCAPE) (pp. 442-446). IEEE.
xxviii. McDermott, C. D., Majdani, F., & Petrovski, A. V. (2018, July). Botnet detection in the internet of things using deep learning approaches. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
xxix. R, Vinayakumar, HB, B. G., Poornachandran, P., & KP, S. (2018). Deep-Net: Deep Neu-ral Network for Cyber Security Use Cases. arXiv preprint arXiv:1812.03519.
xxx. Salvador, P., Nogueira, A., Franca, U., & Valadas, R. (2009, May). Framework for zom-bie detection using neural networks. In 2009 Fourth International Conference on Internet Monitoring and Protection (pp. 14-20). IEEE.
xxxi. Shi, C., Liu, J., Liu, H., & Chen, Y. (2017, July). Smart user authentication through actu-ation of daily activities leveraging WiFi-enabled IoT. In Proceedings of the 18th ACM In-ternational Symposium on Mobile Ad Hoc Networking and Computing (p. 5). ACM.
xxxii. Shi, H., Dong, J., Wang, W., Qian, Y., & Zhang, X. (2017, September). SSGAN: secure steganography based on generative adversarial networks. In Pacific Rim Conference on Multimedia (pp. 534-544). Springer, Cham.
xxxiii. Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., ... & Murata, M. (2017, May). Malicious URL sequence detection using event de-noising convolution-al neural network. In 2017 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.
xxxiv. Siegelmann, H. T., & Sontag, E. D. (1992, July). On the computational power of neural nets. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 440-449). ACM.
xxxv. Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal, 56(1), 136-154.
xxxvi. Sun, J., Wang, X., Xiong, N., & Shao, J. (2018). Learning sparse representation with var-iational auto-encoder for anomaly detection. IEEE Access, 6, 33353-33361
xxxvii. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the in-ception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

xxxviii. Torres, P., Catania, C., Garcia, S., & Garino, C. G. (2016, June). An analysis of recurrent neural networks for botnet detection behavior. In 2016 IEEE biennial congress of Argen-tina (ARGENCON) (pp. 1-6). IEEE

xxxix. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017, March). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In Workshops at the ThirtyFirst AAAI Conference on Artificial Intelligence.

xl. Volkhonskiy, D., Nazarov, I., Borisenko, B., & Burnaev, E. (2017). Steganographic gen-erative adversarial networks. arXiv preprint arXiv:1703.05502.

xli. Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic clas-sification using convolutional neural network for representation learning. In 2017 Inter-national Conference on Information Networking (ICOIN) (pp. 712-717). IEEE.

xlii. Wang, X., & Yiu, S. M. (2016). A multi-task learning model for malware classification with useful file access pattern from API call sequence. arXiv preprint arXiv:1610.05945

xliii. Wu, C. H. (2009). Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. Expert Systems with Applications, 36(3), 4321- 4330.

xliv. Yin, C., Zhu, Y., Liu, S., Fei, J., & Zhang, H. (2018, May). An enhancing framework for botnet detection using generative adversarial networks. In 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD) (pp. 228-234). IEEE.

xlv. Yu, B., Gray, D. L., Pan, J., De Cock, M., & Nascimento, A. C. (2017, November). In-line DGA detection with deep networks. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 683-692). IEEE.

xlvi. Zell, A. (1994). Simulation neuronaler netze (Vol. 1, No. 5.3). Bonn: Addison-Wesley.

xlvii. Zenati, H., Foo, C. S., Lecouat, B., Manek, G., & Chandrasekhar, V. R. (2018). Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222.

xlviii. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycleconsistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).