International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Machine Learning and Deep Learning Techniques for Cybersecurity

DEEKSHA V[1] ,SHREYAS S ALVA [2] ,SUHAS KC [3]

[1,2] U.G. Student, Department of Computer Engineering, CIT Engineering College, Gubbi, Tumkur, India
[3] Assistant Professor, Department of Computer Engineering, CIT Engineering College, Gubbi, Tumkur, India

ABSTRACT:

As the Internet continues to expand, cyber threats are evolving swiftly, posing significant challenges to the cybersecurity landscape. This survey report provides a concise tutorial and reviews essential literature on machine learning (ML) and deep learning (DL) techniques for network-based intrusion detection. Each ML/DL approach is described, with associated research studies reviewed, organized, and summarized according to thermal or temporal relationships. The report also highlights popular network datasets used in ML/DL, addresses the complexities of applying these methods to cybersecurity, and suggests potential directions for future research, underscoring the critical role of data in ML/DL applications.

Keywords: machine learning, deep learning, intrusion detection, cybersecurity.

I. INTRODUCTION :

The Internet's increased integration with social life is changing how people work and study, but it also puts us at greater risk for security issues [2]. Recognizing various network types

Attacks are a serious issue that must be addressed immediately, particularly those that have never been seen before [3]. Cybersecurity is a set of practices and tools designed to protect computers, software, networks, and data from illicit access, alteration, and destruction [4]. A computer security system and a network security system are the two components that make up a network security system [5].

Firewalls, antivirus software, and intrusion detection systems (IDS) make up each of these systems. IDSs assist in identifying, evaluating, and detecting illicit system activities, such as use, duplication, This study focuses exclusively on machine learning and deep learning approaches, rather than describing all of the various methods of network anomaly detection. Nevertheless, aside from oddity Methods based on detection, signatures, and hybrids are shown. In their discussion of technological developments in anomaly detection, Patch and Park [12] point out unresolved issues and difficulties with hybrid intrusion detection systems and anomaly detection systems. However, our survey includes more current studies, while theirs only includes those published between 2002 and 2006. In contrast to Modi et al. [5], this paper is not just about cloud security; it also discusses the use of ML and DL in other domains of intrusion detection. Machine-learning intrusion approaches are the main emphasis of Revathi and Malathi [13]. A wide range of machine-learning techniques for the NSL-KDD intrusion detection system are presented by the authors. dataset, however the scope of their research is limited to misuse detection. On the other hand, this work discusses anomaly detection in addition to misuse detection. modification and devastation [6]. Current instances of security breaches include external intrusions.Security breaches include intrusions from the inside as well as the outside. There are three main types of network analysis for intrusion detection systems (IDSs): hybrid, anomaly-based, and misuse-based, often known as signature-based. Utilizing the signatures to identify known threats of these attacks is the aim of misuse-based detection approaches [7]. When used for known assault kinds, they don't generate many false alarms. However, it is often necessary for administrators to manually change the database rules and 1 signatures [8]. Overused technology makes it impossible to detect novel (zero-day) assaults. Anomaly-based techniques analyze normal network and system behavior and identify anomalies as deviations from it [9]. Their capacity to recognize

II. ML AND DL SIMILARITIES AND DIFFERENCES :

The relationships between machine learning (ML), deep learning (DL), and artificial intelligence (AI) are intricate and overlapping. AI is a field focused on creating technologies that simulate or enhance human-like intelligence[14]. This area includes fields like expert systems, computer vision, robotics, and natural language processing, aiming to simulate human cognition without replicating human thought precisely.

ML is a branch within AI that often overlaps with computational statistics, especially in computer-based predictive modelling, and is closely linked with mathematical optimization, which provides many of its techniques and applications. Unlike data mining, which emphasizes exploratory analysis (often called unsupervised learning), ML also involves creating behavioural profiles to identify anomalies[15]. As Arthur Samuel defined it, ML "enables computers to learn without explicit programming," focusing mainly on classification and regression from training data.

DL, a more recent subfield of ML, is inspired by neural networks resembling the human brain for complex pattern recognition. Initiated by Hinton through the deep belief network (DBN), DL employs unsupervised, layer-by-layer training methods to address complex optimization challenges. Later, LeCun et al. proposed convolutional neural networks to improve training efficiency by using spatial relationships in data.
Key Differences Between ML and DL:

- Data Dependence: DL improves with large datasets, while ML can perform well with smaller data.
- Hardware: DL needs powerful GPUs for matrix operations, unlike most ML methods.
- Feature Engineering: DL automatically extracts high-level features, simplifying feature engineering.
- Problem-Solving: ML breaks down problems, while DL enables end-to-end solutions.
- Execution Time: DL has longer training times but faster testing.

III. DATASET ON NETWORK SECURITY :

Research on computer network security is based on data. To conduct pertinent security research, the right data must be chosen and used reasonably. The training effects of the ML and DL models are also influenced by the dataset's size. There are typically two methods for obtaining data on computer network security: 1) directly and 2) by utilizing an already-existing public dataset. Direct access refers to the use of different techniques for gathering the necessary cyber data directly, such as using software tools like Wireshark or Win Dump to capture network traffic[16]. This method is very focused and appropriate for gathering little or short-term data, but acquisition time and storage expenses will increase for large or long-term data collection.

A. DETECTION OF DARPA INTRUSION

Identifying Data Sets Data Sets [17], collected and maintained by DARPA and AFRL/SNHS, 22 released by MIT Lincoln to evaluate computer network intrusion detection systems The Cyber Systems and Technology Group of 25 Laboratory (formerly the DARPA Intrusion Detection 1 Evaluation Group). The first standard dataset contains a sizable amount of attack and background traffic data. You can get it straight from the website.

B. DATASET KDD CUP 99

Derived from the DARPA 12 1 1998 dataset, the KDD Cup 99 dataset [18] is one of the most widely used training sets. In this dataset, 4,900,000 duplicate assaults have been recorded. There are 22 types of attacks. which are divided into one normal type with the identity of normal and five major groups: The eight distinct attack types include DoS (Denial of Service), R2L (Root to Local), U2R (User to Root), Probe (Probing), and Normal. Each record in the KDD Cup 99 training dataset has 41 fixed 1 feature attributes in addition to a class identification. The remaining qualities are continuous, whereas seven of the 41 fixed feature traits are symbolic in nature.

C. DATASET NSL-KDD

The NSL-KDD dataset is an improved version of the KDD Cup 99 dataset. The NSL-KDD dataset improves. There are certain limitations with the KDD Cup 99 dataset. The KDD 1999 Cup Dataset Intrusion Detection Dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Contest. This approach separates the traits of regular and intrusive connections in order to build network intrusion detectors. Each example in the NSL-KDD dataset has characteristics of a certain type of network data.

D. ADFA DATASET

The ADFA data set, which is a compilation of host-level intrusion detection system data sets commonly used in intrusion detection product testing, was made public by the Australian Defence Academy (ADFA). The dataset's many system calls have been categorized and characterized based on their type. 27 of assault. The data collection includes Windows (ADFA-WD) and Linux (ADFA-LD), two OS platforms that monitor the order 1 of system calls. ADFA-LD records the operating system's invocation for a predetermined period of time. The kernel provides the user space program and communicates with a number of common interfaces. The utilization of system resources, speaking, reading, writing, and manipulating equipment are examples of physical devices to which the user program interface may be constrained. creating a novel process, etc. With user space managing requests and kernel space managing execution, these interfaces serve as a conduit between user and kernel space.

IV. CYBERSECURITY ML AND DL ALGORITHM :

A. MACHINE VECTOR SUPPORT

One of the most precise and dependable machine learning methods is the Support Vector Machine (SVM). Support Vector Classification (SVC) and Support Vector are its two primary components. SVR, or regression. The foundation of SVC is the idea of decision boundaries, which divide instances into discrete groups based on 33 different class values. Both binary and multi-class classifications are supported. Points on one side of the hyperplane are part of one class during one classification, while points on the other side are part of another. In order to make data points that are not linearly separable linearly separable, SVM employs kernel functions to map them into higher-dimensional spaces.that are not linearly separable, the SVM uses the appropriate kernel functions to map them onto higher dimensional spaces, making them separable in those areas.

B. DECISION TREE

In machine learning, a decision tree is a prediction model in which each internal node denotes a test on two specific attributes, each branch denotes the test's outcome, and each leaf node denotes a class three label. With each node standing in for an object, each path 1 for a possible attribute value, and each leaf for the result of the path from the root to the leaf, this structure associates object values with attributes. One output is usually produced by a single decision tree; independent decision trees can be constructed for each of the several outputs.

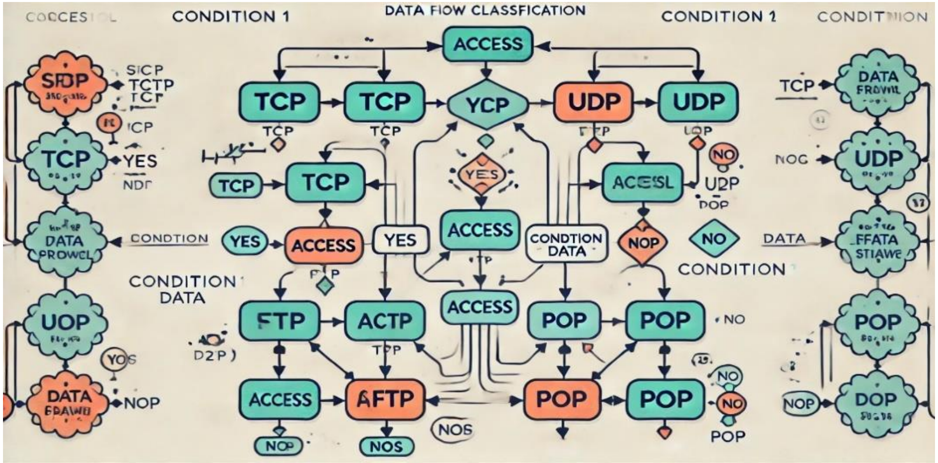

Figure 1. An example decision tree.

C. DEEP NETWORK

A Deep Belief Network (DBN) is a probabilistic generative model that consists of multiple layers of stochastic and hidden variables. Both the DBN and the Boltzmann machine with constraints (RBM) are 1 linked thus, when numerous RBMs are constructed and stacked, many hidden layers can efficiently train data by turning on one RBM for later training stages. An RBM is the special topological structure of a Boltzmann machine (BM). The BM principle was initially presented in statistical physics as a modeling method based on an energy function that may describe the high-order interactions between variables. The symmetric coupled random feedback binary unit neural network, or BM, is composed of a visible layer and many hidden layers. The network node is divided into a visible unit and a hidden unit, and both the visible and hidden units are used to express a random network and environment. The correlation between units is represented by weighting in the learning model. using Deep Belief Nets (DBNs) in their research to detect malicious software. They use PE files from the internet as examples. Through unsupervised learning, DBNs identify many feature layers, which are then input into a feed-forward neural network and tuned to optimize discrimination.

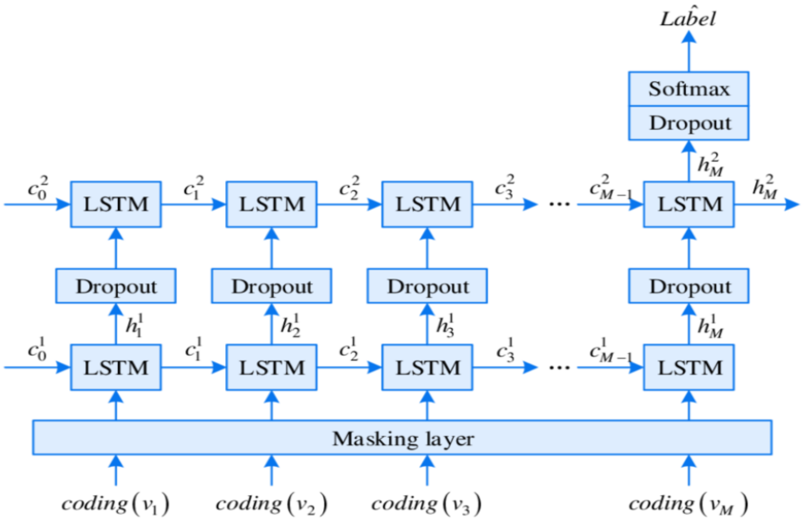

Figure 2. A sample RNN model structure

Compared to feedforward neural networks initialized with random weights, Deep Belief Networks (DBNs) are less prone to overfitting due to their unsupervised pre-training algorithm. This approach enables the training of neural networks with multiple hidden layers. Studies show that DBNs outperform other popular learning techniques, such as SVM, KNN, and decision trees, achieving superior classification performance due to their ability to learn from large amounts of unlabelled data. Although no further details are provided, DBNs demonstrate an approximate accuracy of 96.1%.

To achieve high accuracy with limited labelled data, integrating neural networks with semi-supervised learning can be beneficial. Experiments using the KDD Cup 99 dataset showed that DBNs classified the labelled data after processing the unlabelled data through a Ladder Network, achieving a detection accuracy of 99.17%, comparable to fully supervised learning. However, as with Ding's work, performance metrics beyond accuracy were not calculated. After comparing several DBN architectures and modifying the layer count and hidden units, a four-layer DBN model was developed. Testing on the KDD Cup 99 dataset yielded accuracy, precision, and false alarm rate (FAR) of 93.49%, 92.33%, and 0.76%, respectively. Key challenges addressed included redundant data, large volumes, lengthy training times, and a tendency to get stuck in local optima during intrusion detection.

A DBN-based intrusion detection approach, combined with a probabilistic neural network (PNN), was also proposed. This method first applies the DBN's nonlinear learning ability to transform the data into a low-dimensional form while preserving essential features. The particle swarm optimization algorithm then optimizes the hidden nodes in each layer to enhance learning performance. The reduced-dimensional data is then classified using the PNN. Evaluated on the KDD Cup 99 dataset, this method achieved 99.14% accuracy, 93.25% precision, and a 0.615% FAR. The approach is further refined using a Logistic Regression SoftMax layer, trained over ten epochs to improve the overall network performance.

D. Current Neural Networks

Sequential data analysis is a typical application for recurrent neural networks, or RNNs. Conventional neural networks have no connections between nodes in each layer; instead, layers are fully coupled from input to hidden to output. This conventional structure limits their ability to handle sequence-related issues. In contrast, RNNs, named for their recurrent connections, consider the output of previous steps when calculating the current output, allowing the network to retain information over time. The hidden layer's input includes both the current input layer output and the hidden layer output from the previous step. RNNs theoretically handle sequences of any length, though often only recent states are considered for simplicity. Enhanced RNN models, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, extend this capacity.

In a multi-layer RNN network, the sequential timing properties can be visualized as an expanded network structure, as shown in Fig. 2. These advanced RNN architectures are particularly valuable for sequence-based tasks due to their improved capacity for remembering long-term dependencies.

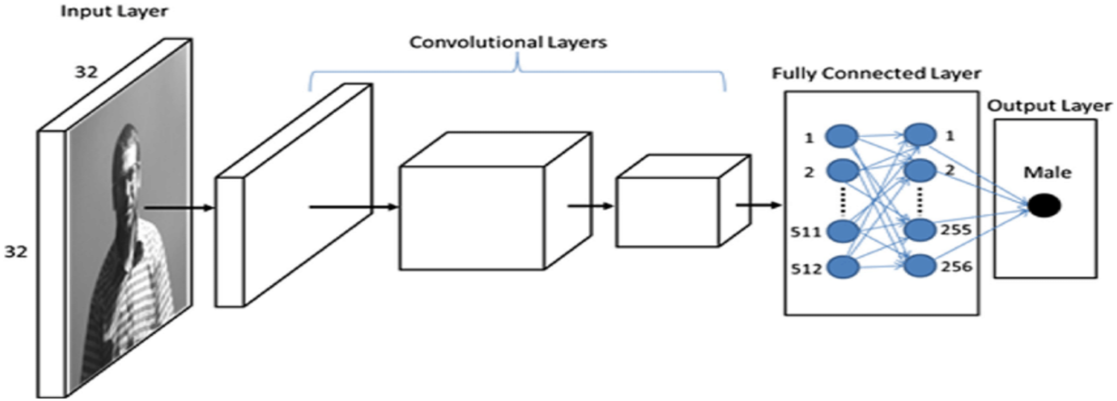

Figure 3. An example CNN model structure.

Both the Contagions-CTU-UNB and CTU-UNB datasets were generated in the same study using distinct malware traffic data. Classification tasks were performed to evaluate the proposed model's performance, achieving precision, recall, and F-score values of 98.44%, 98.40%, and 0.984, respectively. In neural networks, advances are applied to simulate the execution sequence of disassembled malicious binary files. A neural network model incorporating both feedforward and convolutional components is used, blending instruction characteristics with simple convolutional vectorization for hierarchical feature extraction. Features extracted from Portable Executable (PE) header files are used solely for evaluation in this study. Results indicate that this approach outperforms industry standards, including support vector machines and basic feedforward networks, achieving precision and recall of 0.93 with an F1 score of 0.92. Saxe and Berlin introduce an "Exposer" neural network, which uses deep learning with character-level embedding and convolutional neural networks to simultaneously extract features and classify data. This model processes raw short strings (often encountered in security contexts, like potentially malicious URLs, file paths, named pipes, mutexes, and registry keys) as inputs. Exposer fully automates feature design and extraction, outperforming baseline manual methods, with a 0.1% reduction in the false alarm rate and a detection rate of 92%. Additionally, a method for categorizing end-to-end encrypted communication using a one-dimensional convolutional neural network is proposed

V. Discussion and Future Goals :

Our study examines a sizable body of scholarly machine learning and deep learning-based intrusion detection research. These studies highlight important issues in the field and expose a number of discrepancies, namely in the following areas:

1. Limited Benchmark Datasets: Although the same datasets are often used, there is a scarcity of standardized benchmark datasets, and each institution tends to use different sample extraction methods.
2. Biased Results and Inconsistent Evaluation Metrics: Evaluation metrics are often inconsistent, with many studies reporting only accuracy. In contrast, studies using multi-criteria assessments tend to employ different metric combinations, making cross-study comparisons challenging.
3. Limited Focus on Deployment Efficiency: Although real-world deployment efficiency is critical, most research focuses more on algorithmic performance within lab environments rather than on practical deployment, leaving time complexity and detection efficiency underexplored in live networks.

Research on intrusion detection has a future thanks to the issues and patterns mentioned above:

1. Datasets

Current datasets for intrusion detection have several shortcomings, including outdated information, redundant data, and imbalanced category representation. Inadequate data volume remains a concern, even though data quality may improve with processing. A key objective in intrusion detection research is to develop comprehensive datasets that include substantial data volumes, broad attack type coverage, and balanced sample sizes across categories.

2. Hybrid Techniques

Most hybrid detection approaches incorporate machine learning techniques; however, there is relatively limited research on models that blend machine learning with deep learning for intrusion detection. The success of models like AlphaGo demonstrates the potential of this combination, making it an intriguing area for further investigation.

3. Detection Speed

Reducing detection time and increasing detection speed are essential due to the computational complexity of machine learning and deep learning algorithms. To do this, you can maximize both the

algorithms as well as the hardware that supports them. Improvements to hardware, such parallel computing with several processors, can greatly increase processing speed.

4. Adapting to New Threats

With new network penetration methods constantly emerging, adapting models to fit new data is essential. Improving the model's adaptability is a compelling research area. Transfer learning, which allows models to be fine-tuned with a small set of labeled data, is a practical approach to enhance real-time network detection accuracy by enabling models to stay updated with evolving threats.

VI. CONCLUSION :

This study offers an overview of current research on machine learning and deep learning approaches for network security, with a primary focus on the most recent intrusion detection developments over the previous three years. The best intrusion detection method is still unknown, though. A comparison of different It is challenging to choose a single way as the optimum answer because every method of putting an intrusion detection system into place has different benefits and drawbacks.

Datasets for network intrusion detection are necessary for these systems' training and testing. For ML and DL techniques to work well, representative data is necessary, but producing such datasets is a difficult and time-consuming process. Notable problems with the public datasets available today, like inconsistent and out-of-date data, have seriously impeded research advancement in this area. Training ML and DL models is complicated by the ongoing evolution of network information since these models need to be retrained frequently over long periods of time. Future studies in this field will therefore probably focus on incremental learning strategies and lifelong learning to make sure models continue to be flexible and useful.

VII.REFRENCES :

[1] S. Aftergood, ‘‘Cybersecurity: The cold war online,’’ Nature, vol. 547, no. 7661, pp. 30–31, Jul. 2017.

[2] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, ‘‘Evaluating computer intrusion detection systems: A survey of common practices,’’ ACM Comput. Surv., vol. 48, no. 1, pp. 1–41, 2015.

[3] C. N. Modi and K. Acha, ‘‘Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: A comprehensive review,’’ J. Supercomput., vol. 73, no. 3, pp. 1192–1234, 2017.

[4] E. Viegas, A. O. Santin, A. França, R. Jasinski, V. A. Pedroni, and L. S. Oliveira, ‘‘Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems,’’ IEEE Trans. Comput., vol. 66, no. 1, pp. 163–177, Jan. 2017.

[5] A. Patcha and J.-M. Park, ‘‘An overview of anomaly detection techniques: Existing solutions and latest technological trends,’’ Comput. Netw., vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[6] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, ‘‘A survey of intrusion detection techniques in Cloud,’’ J. Netw. Comput. Appl., vol. 36, no. 1, pp. 42–57, 2013.

[7] S. Revathi and A. Malathi, ‘‘A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,’’ in Proc. Int. J. Eng. Res. Technol., 2013, pp. 1848–1853.

[8] D. Sahoo, C. Liu, and S. C. H. Hoi. (2017). ‘‘Malicious URL detection using machine learning: A survey.’’ [Online]. Available: https://arxiv.org/abs/1701.07179

[9] A. L. Buczak and E. Guven, ‘‘A survey of data mining and machine learning methods for cyber security intrusion detection,’’ IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[10] M. Soni, M. Ahirwa, and S. Agrawal, ‘‘A survey on intrusion detection techniques in MANET,’’ in Proc. Int. Conf. Comput. Intell. Commun. Netw., 2016, pp. 1027–1032.

[11] R. G. Smith and J. Eckroth, ‘‘Building AI applications: Yesterday, today, and tomorrow,’’ AI Mag., vol. 38, no. 1, pp. 6–22, 2017.

[12] P. Louridas and C. Ebert, ‘‘Machine learning,’’ IEEE Softw., vol. 33, no. 5, pp. 110–115, Sep./Oct. 2016.

[13] M. I. Jordan and T. M. Mitchell, ‘‘Machine learning: Trends, perspectives, and prospects,’’ Science, vol. 349, no. 6245, pp. 255–260, 2015.

[14] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, pp. 436–444, May 2015.

[15] G. E. Hinton, ‘‘Deep belief networks,’’ Scholarpedia, vol. 4, no. 5, p. 5947, 2009.

[16]. Deng, J.; Zhang, Z.; Marchi, E.; Schuller, B. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2–5 September 2013; pp. 511–516.

[17]. Hinton, G.E. A practical guide to training restricted Boltzmann machines. In Neural Networks: Tricks of the Trade; Springer: Berlin, Germany, 2012; pp. 599–619.

[18]. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. Neural Comput. 2006, 18, 1527–1554. [CrossRef] [PubMed]