Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

# Cybersecurity Deep: Approaches, Attacks Dataset, and Comparative Study

**Kousik Barik, Sanjay Misra, Karabi Konar, Luis Fernandez-Sanz & Murat Koyuncu**

**To cite this article:** Kousik Barik, Sanjay Misra, Karabi Konar, Luis Fernandez-Sanz & Murat Koyuncu (2022) Cybersecurity Deep: Approaches, Attacks Dataset, and Comparative Study, Applied Artificial Intelligence, 36:1, 2055399, DOI: 10.1080/08839514.2022.2055399

**To link to this article:** https://doi.org/10.1080/08839514.2022.2055399



Published online: 25 Mar 2022.

APPLIED ARTIFICIAL INTELLIGENCE
2022, VOL. 36, NO. 1, e2055399 (2710 pages)
https://doi.org/10.1080/08839514.2022.2055399

Taylor & Francis
Taylor & Francis Group

REVIEW



# Cybersecurity Deep: Approaches, Attacks Dataset, and Comparative Study

Kousik Barik[a], Sanjay Misra[b], Karabi Konar[c], Luis Fernandez-Sanz[a], and Murat Koyuncu[d]

[a]Department of Computer Science, University of Alcala, Madrid Spain; [b]Department of Computer Science and Communication, Ostfold University College, Halden, Norway; [c]JIS Institute of Advanced Studies & Research, JIS University, Kolkata, India; [d]Department of Computer Engineering, Atilim University, Turkey

**ABSTRACT**

Cyber attacks are increasing rapidly due to advanced digital technologies used by hackers. In addition, cybercriminals are conducting cyber attacks, making cyber security a rapidly growing field. Although machine learning techniques worked well in solving large-scale cybersecurity problems, an emerging concept of deep learning (DL) that caught on during this period caused information security specialists to improvise the result. The deep learning techniques analyzed in this study are convolution neural networks, recurrent neural networks, and deep neural networks in the context of cybersecurity.A framework is proposed, and a real-time laboratory setup is performed to capture network packets and examine this captured data using various DL techniques. A comparable interpretation is presented under the DL techniques with essential parameters, particularly accuracy, false alarm rate, precision, and detection rate. The DL techniques experimental output projects improvise the performance of various real-time cybersecurity applications on a real-time dataset. CNN model provides the highest accuracy of 98.64% with a precision of 98% with binary class. The RNN model offers the second-highest accuracy of 97.75%. CNN model provides the highest accuracy of 98.42 with multiclass class. The study shows that DL techniques can be effectively used in cybersecurity applications. Future research areas are being elaborated, including the potential research topics to improve several DL methodologies for cybersecurity applications.

**ARTICLE HISTORY**
Received 23 January 2022
Accepted 16 March 2022

## Introduction

Internet usage increased significantly during this pandemic, with sizable interconnected networks facing multiple threats. As a result, there are multiple security threats in cyberspace. Various security organizations worldwide continue to develop innovative techniques to protect peripherals and sensitive

**CONTACT** Sanjay Misra sanjay.misra@covenantuniversity.edu.ng Ostfold University College, Halden, Norway

This article has been republished with a minor change. This change do not impact on the academic content of the article.

data from cyberattacks. Broad security practices include network-based security systems (Zhengibing, Zhitang, and Junqi 2008) and host-based (Hu 2010) that protect cornered peripherals from illegal intrusion.

These systems consist of multiple devices combined, primarily firewalls, intrusion detection systems (IDS), threat protection, simple control over system practices, and a flag boost based on configured detection priority. Intrusion detection plays an essential role in the configured detection priority. Intrusion detection plays an essential role in information security and helps detect illegal access, changes, and destruction of information systems (Mukkamala, Sung, and Abraham). IDS are generally divided into signature-based, statistical anomaly-based, and combined approaches. Signature-based detection uses predefined signatures of abuse activity to classify intrusion attempts. Statistical anomaly-based detection incorporates natural sequences and identifies suspicious activity based on deviations from routine lines.

On the other hand, the combined approach of detection techniques practiced abuse detection and anomaly detection techniques (Alazab et al. 2010). Many vendors like Microsoft, Checkpoint, Symantec, McAfee, Kaspersky, Symantec, Microsoft McAfee have advanced anti-malware, virus, and threat protection products to protect networks and user data from attacks. Additionally, these vendors typically use signature-based approaches to identify malware. Ransomware attacks (Mcintosh et al. 2019), zero-day attacks (Alazab et al. 2011), unauthorized access (Shenfiled. Dey, and Ayesh 2018), denial of service (DoS) (Larson 2016), data breaches (Low 2017), phishing (Binks 2019), social engineering (Krombholz et al. 2015), etc. common nowadays. These security incidents or cybercrimes intensely impact businesses and people, causing disruption and overwhelming business and financial losses. These security incidents or cybercrime intensely impact businesses and people, causing disruption and overwhelming business and financial losses.

DL algorithms have found an essential role in solving complex problems. DL can be classified as various multi-layered ML techniques that capture general notions of complex, vast amounts of data. DL offers several cybersecurity companies modernization of security systems at an optimal cost. The popularity trend is shown in Figure 1, determined from Google Trends from 2019 to 10$^{th}$ January 2022.

Researchers seek complete and accurate data to advance their approaches. However, obtaining the correct, valuable data is a significant difficulty. The contributions of the work are summarized as follows:

(1) A framework for cybersecurity based on deep learning has been proposed.
(2) Real-time data has been collected using a lab setup, and then the proposed methodology for the evaluation process has been insulated.

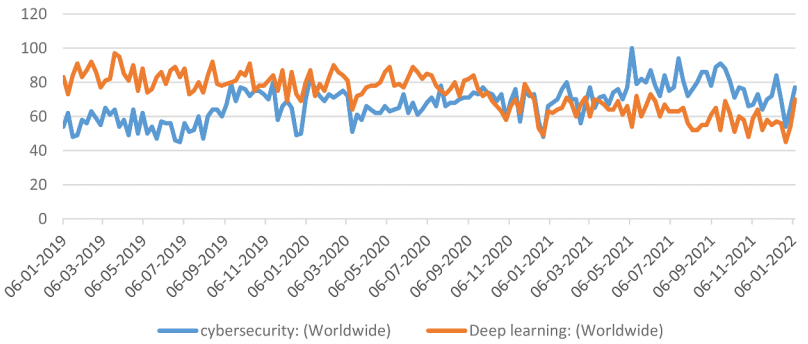



**Figure 1.** Popularity score of "cyber security" and "deep learning" worldwide from 2019 to 10th January 2022, the x-axis represents time strap, and the y-axis represents popularity score.

(3) A detailed evaluation with live data is carried out using three different deep learning techniques. It shows how deep learning techniques can be used effectively in cybersecurity applications.
(4) The key areas of deep learning in the cybersecurity domain and open gaps are explored; new studies can be targeted.

The remaining paper is structured as follows. Section 2 outlines the relevant works, different deep learning approaches, and common types of cybersecurity attacks.. Section 3 introduces the proposed deep learning-based cybersecurity framework. Section 4 discusses the real-time lab setup, the log acquisition process, the methodology employed in the experimentation process, and performance analysis using DL approaches. Section 5 presents the future scope along with open research challenges. Finally, Section 6 confers the conclusion.

## Literature Review

### *Related Works*

Several relevant studies are consistent with DL methods for detecting cybersecurity attacks . We, therefore, classify the studies listed in Table 1 assessed using the systems used:

### *DL Techniques*

The case study defines DL techniques for cybersecurity attack detection systems.

**Table 1.** Study on cybersecurity intrusion detection systems using deep learning.

| Study | Year | DL | Interpretation of DL | Data Set Employed |
|---|---|---|---|---|
| (Haddadi et al. 2010) | 2010 | Yes | No | Partial |
| (Amiri et al. 2011) | 2011 | No | No | Yes |
| (Koc, Mazzuchi, and Sarkani 2012) | 2012 | No | No | Yes |
| Jamdagni et al.[15] | 2013 | No | No | Partial |
| Kuang et al. [16] | 2014 | No | No | Partial |
| (Nadiammai and Hemalatha 2014) | 2014 | No | No | Yes |
| (Buczak and Guven 2015) | 2015 | No | No | Yes |
| (Singh, Kumar, and Singla 2015) | 2015 | Yes | No | Yes |
| (Jabez and Muthukumar 2015) | 2015 | Yes | No | Yes |
| (Milenkoski et al. 2015) | 2015 | No | No | No |
| (Folino and Sabatino 2016) | 2016 | No | No | Partial |
| (Van and Thinh 2017) | 2017 | Yes | No | Partial |
| (Xin et al. 2018) | 2018 | Yes | No | Partial |
| (Loukas et al. 2019) | 2019 | No | No | Partial |
| (Mahdavifar and Ghorbani 2019) | 2019 | Yes | No | No |
| (Costa et al. 2019) | 2019 | No | No | Partial |
| (Ferrag et al. 2020) | 2020 | Yes | Yes | Yes |
| (Wu et al. 2020) | 2020 | Yes | No | Yes |
| Our Study | - | Yes | Yes | Yes |

DL: Deep Learning; IDL: Interpretation of Deep Learning; Dset: Data Sets.

### *Interpretation of DL Techniques*

It shows whether the application evaluates DL techniques for cybersecurity attack detection systems. Data related to IDS shows the work on converging the data associated with cybersecurity attack detection systems.

(Milenkoski et al. 2015) presented the typical applications of cybersecurity IDS by examining existing systems correlated with model assessment criteria. (Loukas et al. 2019) presented a study on cyber-physical IDS strategies for vehicles. (Mahdavifar and Ghorbani 2019) presented a survey on deep learning cybersecurity applications; The highlighted studies focus on Android-based malware detection and analysis. (Ferrage et al. 2020) presented a comprehensive overview of cybersecurity attack detection using DL approaches. In addition to public data sets, various deep learning models are discussed.

Furthermore, two data sets are evaluated with DL methods. (Ring et al. 2019) showed a reflection of intrusion detection data. The study maps different data sets and recognizes special features. Our study focuses on DL techniques intended for cybersecurity IDS on real-time datasets.

However, this work (Mahdavifar and Ghorbani 2019), (Wu et al. 2020), (Yang et al. 2018a) does not provide a comprehensive analysis with DL methods applied to data. Therefore, our study evaluates deep learning approaches using a real-time data set generated by the proposed lab facility instead of a public dataset.

### Deep Learning Approaches

Deep Learning (DL) describes a family of artificial intelligence (AI) derived from artificial neural networks (ANN) (Sarker et al. 2020). However, DL's main interest in traditional machine learning (ML) techniques continues with increasing mass production approaching numerous cases, especially discovering enormous amounts of data (Yang et al. 2018a). This section explains the DL techniques related to cybersecurity IDS. Three deep learning methods for cybersecurity IDS are used in this study, namely (a) CNN, (b) RNN, (c) DNN.

### Convolution Neural Network (CNN)

A convolutional neural network selections elements at a higher resolution and thus translates them into superior compound elements, shown in Figure 2. (Yang et al. 2018b) proposed a feature training system to categorize malicious traffic. CNN is used to study categorizations, and inexperienced traffic flow is directed in pictures through CNN. CNN rates suspicious traffic for image analysis based on the USTC-TRC2016 dataset.

(Chowdhury et al. 2017) suggested using techniques for IDS. The CNN is prepared, successive outputs of CNN are collected and provided as input within the SVM (Support Vector Machine) and KNN (k-nearest neighbor) intrusion detection techn

iques using KDD 99 and NSL-KDD records. (Guo, Wang, and Wei 2018) proposed an analysis of malware detection using CNN. The CNN technology mentioned above includes two folding layers, pooling layers, and inner product layers. They downloaded malicious applications used in this work from the Virus Share website. (Khan et al. 2019) proposed a CNN model that automatically extracts the penetration example. The author used KDD99 datasets to test the accuracy of intrusion detection.

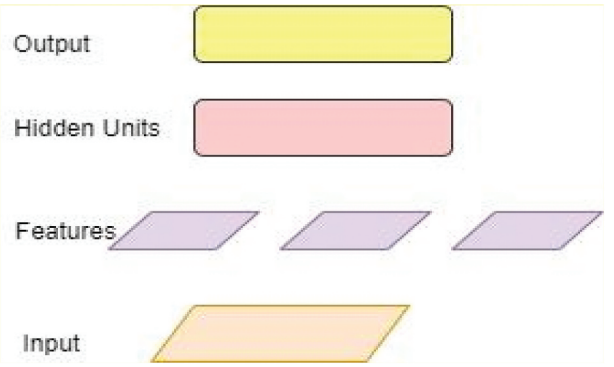



**Figure 2.** Convolutional neural networks.

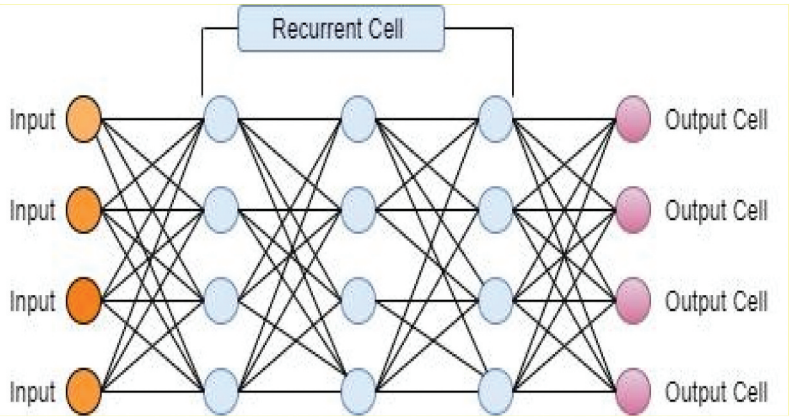



**Figure 3.** Recurrent neural network.

### *Recurrent Neural Network (RNN)*

It's a neural network; the combined diagram contains at least one cycle, shown in Figure 3. (Kim et al. 2016b) proposed a framework using the KDD Cup 1999 dataset for a recurrent neural model for intruder detection. They showed 98.8% among the total attack patterns. (Yin et al. 2017) presented a recurrent neural network integration into an IDS system. They used the NSL-KDD data set and assessed performance based on accuracy, false-positive rate, and true positive rate. The study also highlighted the benefits of using RNN for IDS. (Kim et al. 2016a) presented the LSTM recurrent neural network method for intrusion detection data. They achieved an accuracy of 93.85% and a FAR of 1.62%. (Brown et al. 2018) presented log anomaly detection using RNN. Using the Los Alamos National Laboratory (LANL) cybersecurity dataset, they evaluated model performance.

### *Deep Neural Networks (DNN)*

It features multi-layered perceptions (MLP) with multiple layers and a class of feed-forward artificial neural networks, shown in Figure 4. (Tang et al. 2016), presented an intrusion detection system using CNN and other DL methods in software-defined networks. They used the NSL-KDD data set, and the experimental results found that the learning rate of 0.001 is achieved more effectively than others. (Zhang et al. 2019) presented a deep adversarial and statistical learning method to detect network intrusions. They used two main components, discriminator and generator, in the proposed systems. (Wu and Guo 2019) presented a deep learning model for detecting intruders in an extensive . The models are evaluated using NSL-KDD and UNSW-NB15. Finally, (Rezvy et al. 2018) proposed an intruder detection and classification model using a deep autoencoder algorithm for dense neural networks. They used the NSL-KDD record. They reported an overall recognition accuracy of 99.3%.

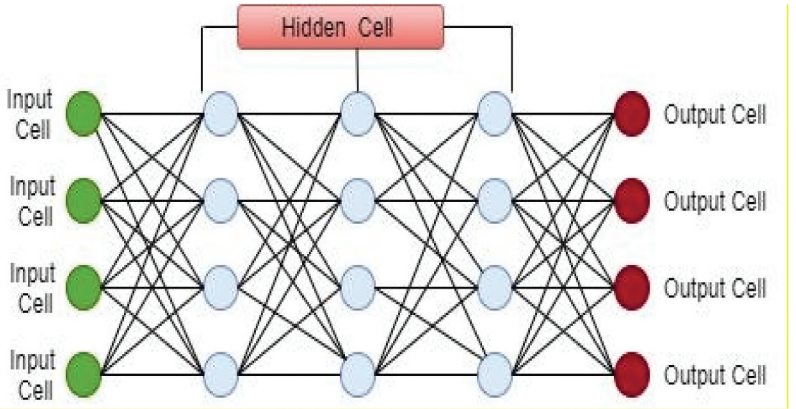



**Figure 4.** Deep neural network.

### *Cybersecurity Attacks*

Sophisticated attacks penetrate cybersecurity operations. This section examined several publications on detecting cybersecurity crimes using deep learning thought, examined some types of attacks, and discussed the variety of intruders, including targets. A summary of common cyber attacks is presented in Figure 5.

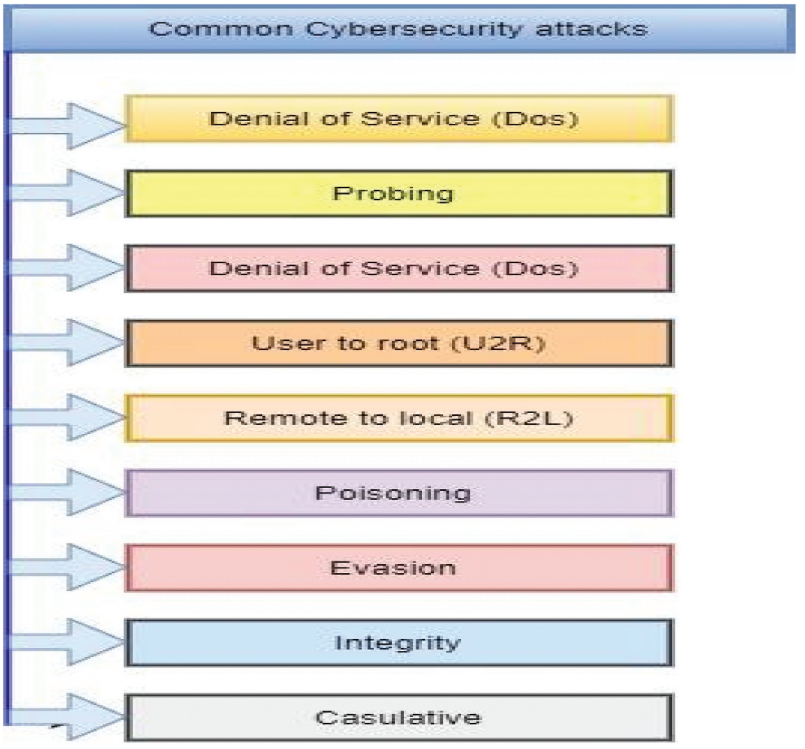



**Figure 5.** Common cyberattacks.

### Attacker Types

Attacks can be divided into three types. First, the attacker has no prior knowledge of systems or deep learning models and no knowledge of the black box attack. Second, the gray box test attackers understand some of this system's information and design elements and have credible information. Third, the attacker has extensive details about the white-box model, which only occurs in the most critical case. Finally, an attack that transfers a focused system to a neural network begins with a misclassification called an integrity violation.

The motive varies with the following analysis. When an attack pushes a focused system onto the neural network, a misclassification known as an integrity violation begins; if the attack is targeted, the operation will not appear and run for some time, treated as an availability violation. On the other hand, when an attacker attempts to negotiate private information, treated as a privacy invasion, this criminal carries out this attack in two main forms: targeted and random attacks. First, the opponent tries to get an inaccurate return on the given attack with a specific part of the training session. Second, the intruder focuses on the practice pattern in an accidental attack.

### Denial of Service Attack (Dos)

It is processed while transmitting a significant volume of traffic in a special technique to the selected recipient; Target users do not have access to the operation of the network. The ultimate goal is to permanently or temporarily suspend or terminate the service (Diro and Chilamkurti 2018). It is handled by transferring a significant volume of traffic to the designated beneficiary; they are no longer entitled to network operation with the ultimate goal of permanently or temporarily suspending or terminating the service (Diro and Chilamkurti 2018).

### Probing

The attackers examine the networks and effectively obtain the information and data (Radford, Metz, and Soumithchintala 2015).

### User to Root Attack

The attacker's track system and regular user account are efficient . Significantly, the identifications are recorded, and confidential information may suffer (Xiong and Yu 2018).

### Remote to Local Attack

The attackers exploit the operation by exploiting the abuse of system communication and execution through the vulnerabilities previously introduced into the process. Remote abuse seems easier to stop, while local attacks are difficult to identify (Mnih et al. 2016).

### Adversarial Attacks

It demands that topics related to DL in privacy statements are appropriate. For example, (Mahloujifar, Diochnos, and Mahmoody 2019) examined the bias attack technique to overcome the obstacle of improving comfort in a realistic environment.

### Poisoning and Evasion Attacks

Poison attacks are performed during each DL preparation phase. Then the attacker interpolates the infection within the preparation samples, reducing the prediction efficiency of the DL technique. An evasive attack targets DL's prediction process. (Jiang et al. 2020) used the Particle Swarm Optimization (PSO) method to combat the virus and focused on this preparatory phase. On the other hand, the poison abuse and the intervention phase point to mysterious attacks.

### Integrity Attacks

When altered or misrepresented, they converge that the information is functional. The attacker largely accompanies this attack by encrypting company-owned elements and accusing the decryption of massive financial fraud.

### Causative Attacks

It is performed while focusing on the decision-making technique to develop a misleading classification of neural networks. (Sihag and Tajer 2020) recommended a method to evaluate the protected framework to detect abuse and isolate the neural network design to overcome this obstacle. The publicly available cybersecurity datasets are summarized in Table 2.

This section examined related work, different DL approaches, and common cybersecurity attacks.

**Table 2.** Cyber security-related public datasets.

| Public dataset | Year of release | Reference | Citations as of 14/08/2021 |
|---|---|---|---|
| DARPA dataset | 1998 | (Lippmann et al. 2000) | 1258 |
| NSL-KDD dataset | 2009 | (Tavallaee et al. 2009) | 3019 |
| MAWI dataset | 2011 | (Fonttugne et al. 2020) | 274 |
| ISCX dataset | 2012 | (Shiravi et al. 2012) | 852 |
| ADFA2013 dataset | 2013 | (Creech and Hu 2013) | 264 |
| CTU-13 dataset | 2013 | (Garcia et al. 2014) | 473 |
| TWENTE dataset | 2014 | (Shoaib et al. 2014) | 406 |
| ICS cyber-attack dataset | 2015 | (Pan, Morris, and Adhikari 2015 | 281 |
| CAIDAs dataset | 2017 | (Jonker et al. 2017) | 76 |
| CICDS2017 dataset | 2017 | (Sharafaldin, Lashkari, and Ghorbani 2018) | 985 |
| Bot-Iot 2018 dataset | 2018 | (Koroniotis et al. 2019) | 259 |
| CIRA-CIC-DoHBrw-2020 dataset | 2020 | (Banadaki 2020) | N/A |

## Proposed Deep Learning-based Cybersecurity Framework

This section illustrates a standard DL framework structure for leveraging cybersecurity. The design is deemed to be comprehensive to address various cybersecurity challenges. Furthermore, the facility will be as in-depth as probable to handle multiple cybersecurity challenges. The visual representation of the designed system is demonstrated in Figure 6. The functional area emphasizes the selection of data sources concerning the proposed framework. The general structure consists of four main elements.

### *Analysis*

The workflow for this structure starts with examining different types of file formats in static and dynamic mode. The structured workflow starts with a static or dynamic examination of network traffic, flow logs, and other log files such as web and cloud. The input file is decompressed in the static analysis phase to extract identical features based on the predefined ruleset. Packet streams are recorded using the defined filter patterns in the dynamic analysis phase. It also practices an on-demand control protocol that allows the filtering rules to be corrected as network requirements change.

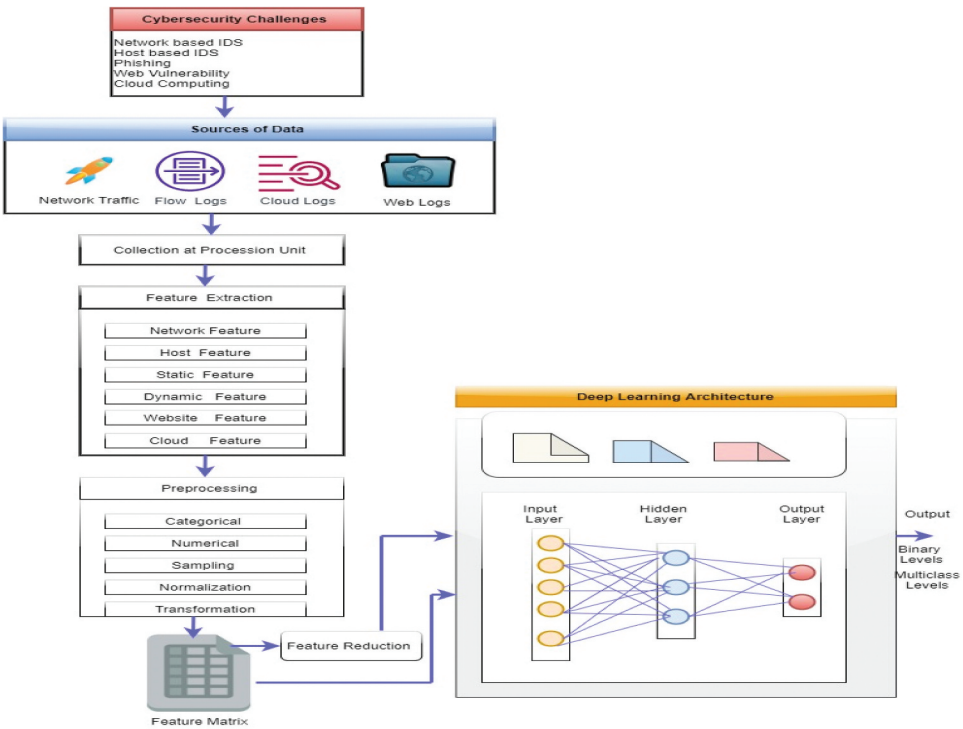



**Figure 6.** The framework of DL for cybersecurity applications.

### *Feature Extraction*

In this module, the static functions, comprising web URL functions, raw log data, and access control-related details, are removed from these accessible resources. In addition, the dynamic functions are extracted from the collected log files, including system sequences, system resources, traffic flow, registry keys, file details, and domain-based. The network feature extractor is a flow-based extractor that can derive network traffic characteristics from various files like Libcap, Wincap, PCAPng, Npcap.

### *Pre-processing*

Input features can be reduced to sub-dimensional subspaces using the random projection system or principal components analysis, which inputs the DN model based on the dimension of this feature matrix. Based on these generated features, we need to normalize the nominal values to the numeric values within the target system, similar to the [1,0] range. There is no need to reduce the dimensionality as the DN is roughened to achieve this naturally. A DL structure framework model should acquire the high-level features of the low-level layer by layer. Therefore, feature conversion before training the DN architecture would not be prudent and would eliminate DL techniques.3.4. DL Classifier.

DL exercises the final feature matrix as input and is trained to use the greedy layer-by-layer learning technique (Tavallaee et al. 2009). Thus, the DL can be implemented by a CNN, DNN, DBN, etc., depending on the input feature matrix. While the DL is shown in the frame, it can be performed using ANN, RNN, CNN, DBN, DNN based on the nature of the input feature matrix. For example, CNN has been very effective at classifying images, and RNN is adapted to processing input sequences.

## Data Collection and Implementation

Create an unbiased, real-time record of intruders that combines a variety of real-world attacks. This section defines the contemporaries of real-world network attack records, including infiltration, brute force, DDoS, and SQL injection records. The experiment is performed in Anaconda Environment with Python 3.8 version using Tensor Flow, Keras. The proposed infrastructure includes four elements: (4.1) proposed setup, (4.2) methodology, (4.3) feature selection, (4.4) processing, (4.5) performance evaluation, (4.6) outcome and discussion, described in Figure 8.

### Proposed Lab Setup

The proposed setup consists of a router, switch, two laptops, and a server. Two virtual machines are installed on each laptop as virtual servers running the Ubuntu operating system, as shown in Figure 7. Attacks are generated both inside and outside the network. Snort 3.0 version, released in January 2021, is used as IDS software, Wireshark tool to capture packets, Scythe, Netssi2 tools to generate attack scenarios, KIWI Syslog server, and MySQL database to store the logs used. In this setup, the Snort analyzer summarizes and recognizes the packets. The analytics engine is an integral part of Snort. Attack simulations are generated from both internal and external networks.

### Proposed Methodology

The methodology adopted in this study is illustrated in Figure 8. After capturing these logs, they are stored in the MySQL database. The dataset has been split into training and testing. 70% has been used as training, and 30% has been used as testing. In the training phase, logs have been classified into begin or attack. We processed the records and followed them by normalization. Different deep learning approaches are applied to the training dataset. Similarly, processing and normalization have been done on testing data. A detection model is developed concerning the received inputs of the preparation and test dataset; an attack detection model is developed.

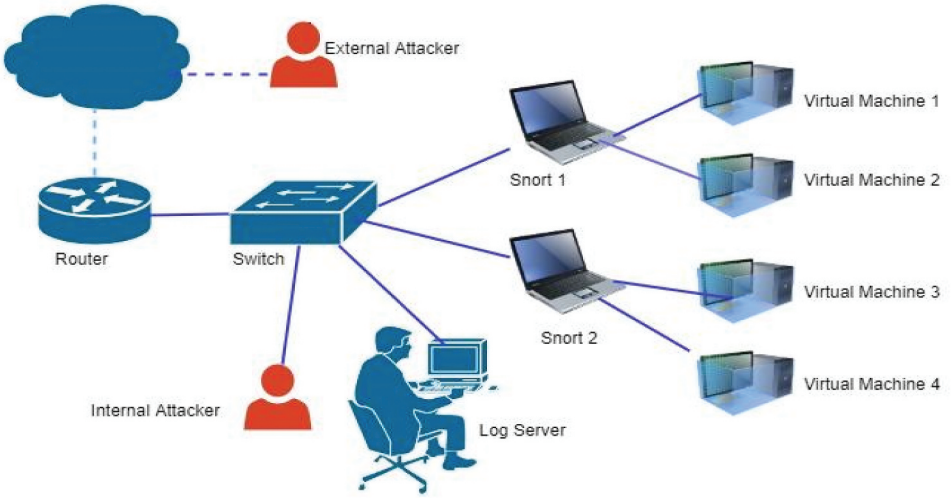



**Figure 7.** Proposed lab setup.

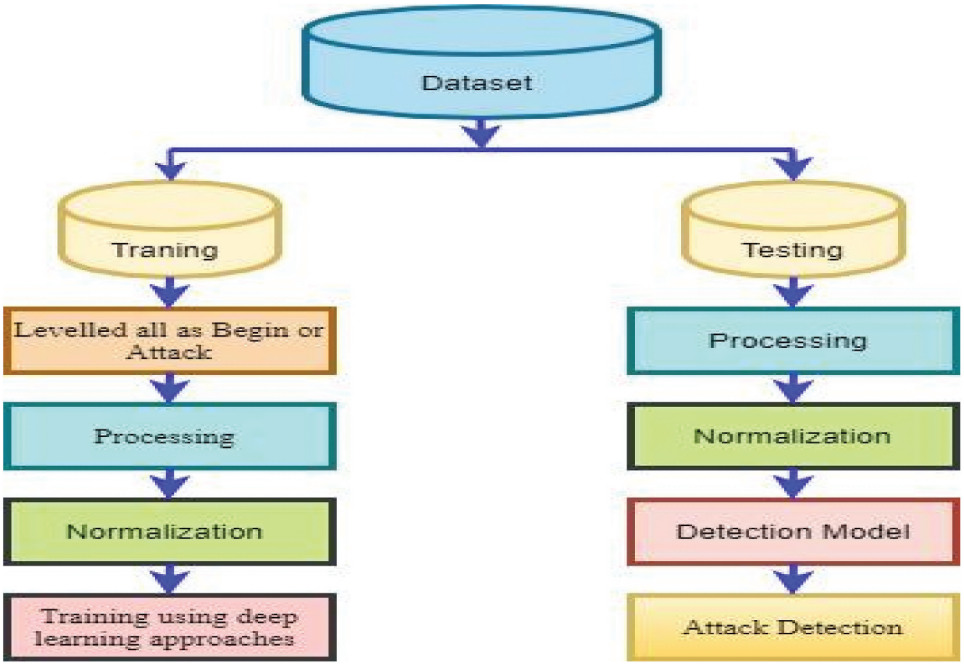



**Figure 8.** The methodology adopted in experimentation.

### *Feature Selection*

The information gain feature mechanism is used to classify the data set. Logs are collected based on 17 characteristics. Traffic is classified into seven categories, shown in Table 3.

### *Processing*

The dataset contains 10,88,365 rows on four files, each row having 17 features. In addition, we parsed and removed column headers repeated in some data files. As a result, about 9768 samples dropped during the data clean-up process. Table 4 represents the summary of the dataset used for experiments.

**Table 3.** Network traffic classification.

| Category | Description |
|---|---|
| Begin | Normal Traffic |
| Infiltration | Attempts to enter and or damage a user's computer |
| Brute Force FTP | Point storage group from extreme false FTP login efforts. |
| Brute Force SSH | Remote logins, command execution, file transfer |
| Brute Force Web | Authentication and discovering hidden content/pages within a web application |
| DDoS | An attack to produce network rescores unavailable to deliberate users. |
| SQL Injection | Web security vulnerability related to the database. |

**Table 4.** Dataset summary.

| Dataset | Traffic Types | No of the Samples dropped | No of Reaming Samples |
|---|---|---|---|
| 07.08.2021.csv | Begin | 4916 | 195326 |
| | Infiltration | 580 | 37125 |
| 08.08.2021.csv | Begin | 1615 | 180216 |
| | Brute Force FTP Brute Force SSH | 3 | 12156 |
| | | 5 | 16896 |
| 09.08.2021.csv | Begin | 2100 | 256106 |
| | Brute Force Web | 6 | 95 |
| | DDoS | 3 | 93126 |
| 10.08.2021.csv | Begin | 531 | 236126 |
| | SQL-Injection | 2 | 189 |
| | DDoS | 7 | 51236 |

**Table 5.** Total number samples for each attack type among all the datasets.

| Traffic Pattern | Total number of samples |
|---|---|
| Begin | 867774 |
| Infiltration | 37125 |
| Brute Force FTP | 12156 |
| Brute Force SSH | 16896 |
| Brute Force Web | 95 |
| DDoS | 144362 |
| SQL-Injection | 189 |

Each of the cleaned datasets contains 17 characteristics, two of which target ports and protocols are treated as categorical using a 1-to-n encoding, and the rest are all numeric. Therefore, Table 5 presents the total traffic statistics samples for a particular type among all datasets; The total number of attack samples is 1078597 and is shown graphically in Figure 9.

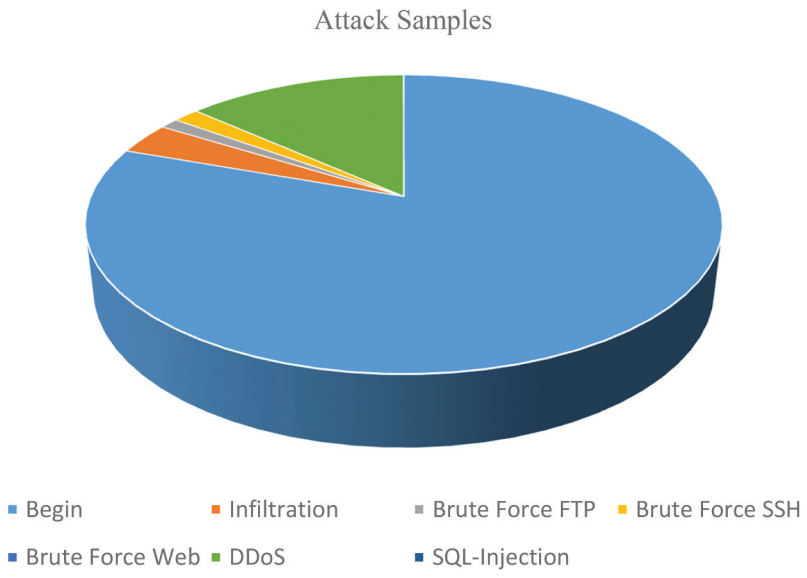



**Figure 9.** Attack samples.

**Table 6.** Summary of evaluation parameters.

| Terminology | Formula |
|---|---|
| Accuracy | $\frac{TP\ attack+TN\ begin}{TP\ attack+TN\ begin+FP\ begin+FN\ attack}$ |
| Precision | $\frac{TP\ attack}{TP\ attack+FP\ begin}$ |
| False Alarm acceptance rate | $\frac{FP\ begin}{TN\ begin+FP\ begin}$ |
| Attack Detection Rate | $\frac{TP\ attack}{TP\ attack+FN\ attack}$ |

## *Performance Evaluation*

Performance evaluations are performed on this dataset to determine the capacity of deep learning approaches to detect cyberattacks and respond within the performance limitations – most critical analysis pointers, including detection, false alarm, precision, and accuracy represented in Table 6.

Where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive. FN denotes False Negative.

## *Results and Discussion*

Deep-learning approaches are utilized on the individual dataset and depicted the performance outcomes such as accuracy, detection rate, and false alarm. Normalizing the numeric features is explored, but the performance variation was statistically tiny to deserve normalizing numeric values for all the experiments. The learning rate used is from 0.01 to 0.8; no. of hidden nodes selected are 15 to 100, batch size 2000, Sigmoid is employed as an activation function. The correlation map of the dataset is presented in Figure 10.

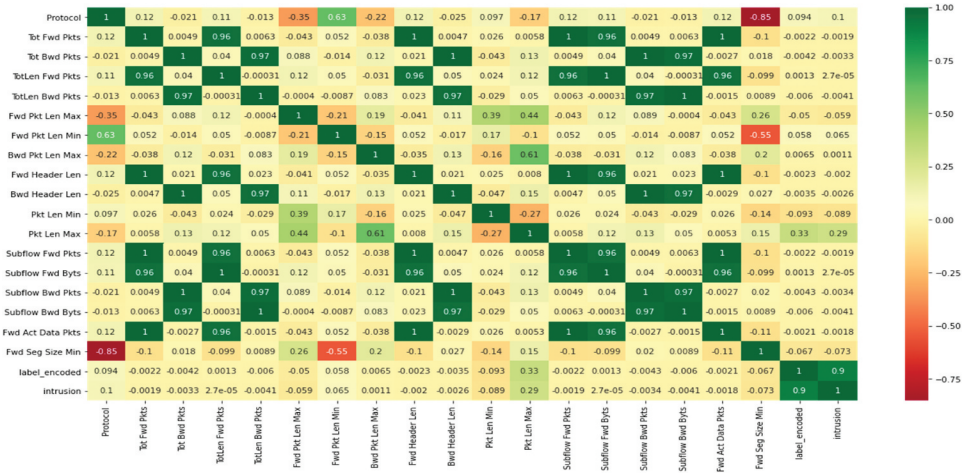



**Figure 10.** Correlation map of the dataset.

The practice of the models over different attack types, detection rate, and normal states, CNN provides the highest true negative rate with 98.8% and highest detection rate of three attack types, namely Infiltration 98.1%, Brute Force FTP 98.7%, and DDoS 98.2%. The RNN gives the highest detection rate of two attack types: Brute Force SSH 97.3% and Brute Force Web 96.5%. The DNN provides the highest detection rate of SQL injection of 98.1%, presented in Table 7.

The ROC Curve (Receiver Operating Characteristic Curve) is presented in Figure 11 with the highest detection rate of the three techniques.

Table 8 shows the accuracy and training time of DL models with various parameters in the dataset. Related to both DBN and RNN networks, CNN obtains a greater accuracy of 98.35%.

Tables 9 and 10 illustrate the accuracy results over 100epochs for Binary-class and Multiclass experiments.

The practice of the models of using Binary-class shows the CNN model provides the most excellent accuracy, about 98.36% with a 98% precision rate. The RNN model offers the second-highest accuracy of 97.75%. The practice of the models using Multiclass shows that the CNN model gives the most excellent accuracy of 98.42%, including a precision rate of 98. The RNN model offers the second-highest accuracy of 97.75%. The comparison of the model is presented in Figure 12.

## Future Scope and Open Research Challenges

Researchers have introduced various methods using DL algorithms to identify, classify and predict the diverse field of cybersecurity. Figure 13 describes the main areas where DL can be used for cybersecurity. First, unnecessary security warnings and comments can indicate how to deal with waste and inaccurate conclusions, a major challenge in deep learning. Then deep learning techniques tend to be uselessly improved when the confidence cases are terrible, namely bad, irrelevant components or insufficient training capacity. Most research results were proposed using the public database. The research should highlight building a real-time setup to validate deep learning approaches so

**Table 7.** The attack detection rate of deep learning models concerning the different attack types and begin.

| Dataset | CNN | RNN | DNN |
|---|---|---|---|
| Begin | 98.8 | 98.2 | 96.3 |
| Infilteration | 98.1 | 97.9 | 96.2 |
| Brute Force FTP | 98.7 | 98.1 | 97.6 |
| Brute Force SSH | 97.1 | 97.3 | 96.9 |
| Brute Force Web | 94.2 | 96.5 | 95.5 |
| DDoS | 98.2 | 97.9 | 96.1 |
| SQL Injection | 97.1 | 97.2 | 98.1 |

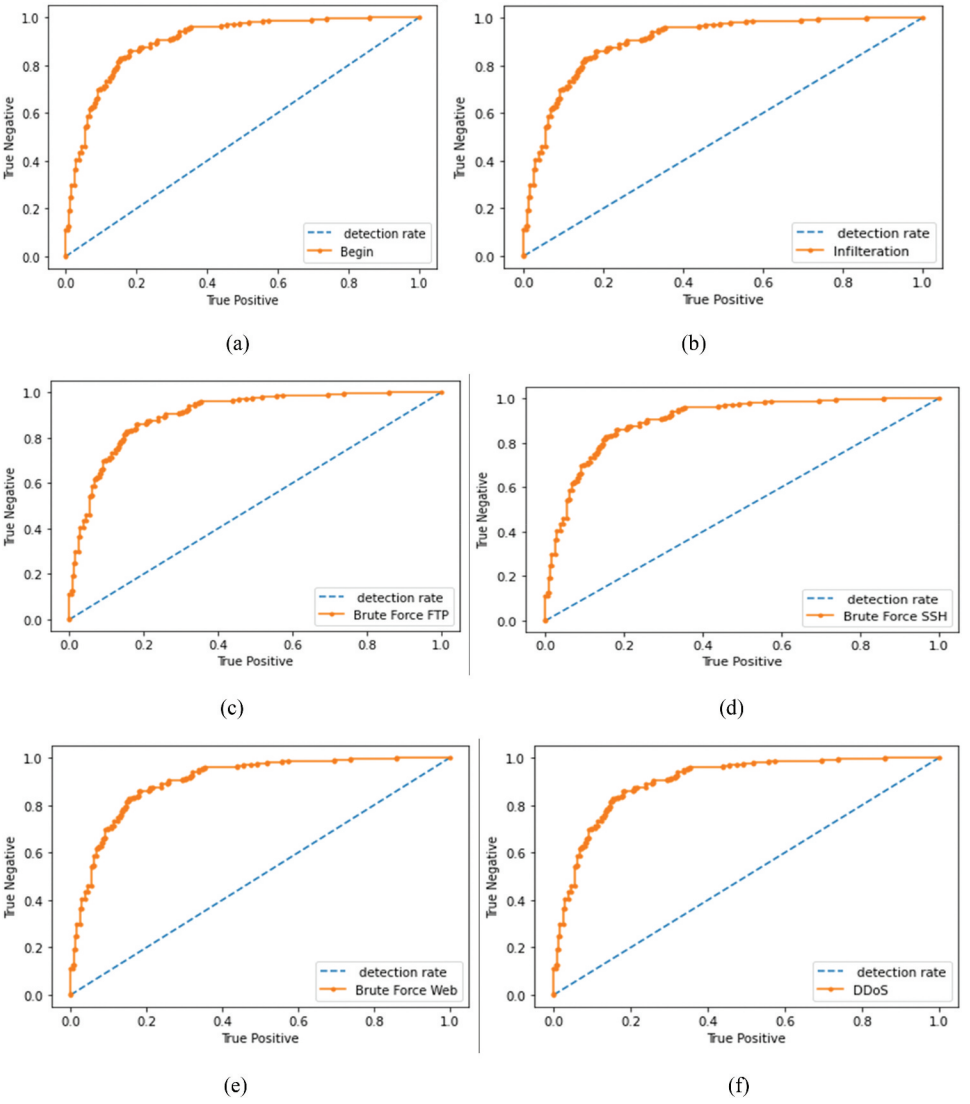
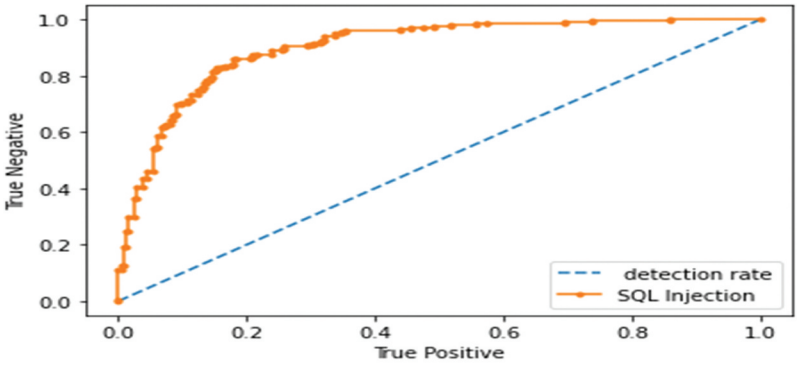



**Figure 11.** Comparison of models

they can counter new types of cybersecurity attacks. Researchers should primarily focus on issues where a criminal uses the DL Technique method to break into the compromised procedure previously acquired using DL Techniques.

Accessing real-time datasets is a challenge. The experiment primarily produces accessible data. The researcher can refine the study to examine various open-source data in the future. Compared to cybersecurity, the DL procedures are associated with higher costs for the corresponding error correction. In addition, the DL methods are linked to black boxes; The advanced principles




(g)

**Figure 11.** (Continued).

**Table 8.** DL techniques' accuracy and training time by various learning rates and hidden nodes.

| Parameters | Accuracy and timings | CNN | RNN | DNN |
|---|---|---|---|---|
| HN = 10 | ACC | 97.235 | 96.85 | 96.245 |
| LR = 0.01 | Time | 38.4 | 41.2 | 26.5 |
| HN = 10 | ACC | 97.153 | 96.825 | 96.235 |
| LR = 0.3 | Time | 35.4 | 36.7 | 24.5 |
| HN = 40 | ACC | 97.325 | 96.725 | 96.125 |
| LR = 0.01 | Time | 90.5 | 96.2 | 79.5 |
| HN = 40 | ACC | 98.205 | 96.625 | 96.452 |
| LR = 0.3 | Time | 91.5 | 97.5 | 80.1 |
| HN = 80 | ACC | 97.805 | 96.725 | 96.535 |
| LR = 0.1 | Time | 186.5 | 191.5 | 161.5 |
| HN = 80 | ACC | 97.805 | 96.725 | 96.535 |
| LR = 0.5 | Time | 195.5 | 193.5 | 164.5 |
| HN = 100 | ACC | 98.25 | 97.125 | 96.895 |
| LR = 0.1 | Time | 375.2 | 360.1 | 335.3 |
| HN = 100 | ACC | 98.35 | 97.82 | 96.92 |
| LR = 0.5 | Time | 410.1 | 375.3 | 365.6 |

**Table 9.** Performance results of the dataset using binary-class.

| Model | Accuracy | Precision |
|---|---|---|
| CNN | 98.36 | 98 |
| RNN | 97.85 | 97 |
| DNN | 96.63 | 96 |

**Table 10.** Performance results of the dataset using multi-class.

| Model | Accuracy | Precision |
|---|---|---|
| CNN | 98.42 | 98 |
| RNN | 97.75 | 97 |
| DNN | 96.81 | 96 |

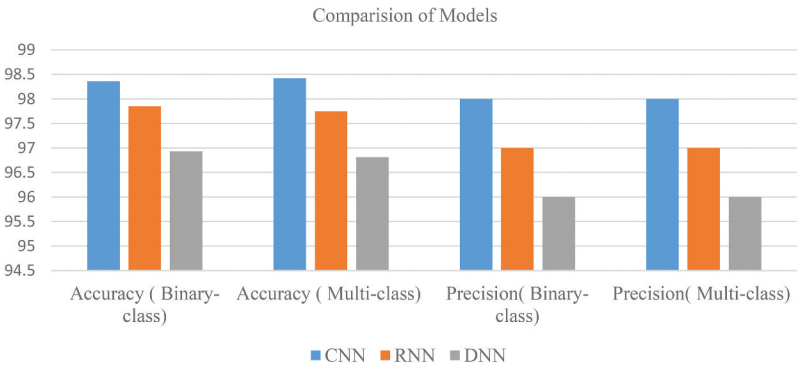



**Figure 12.** Comparison of models.

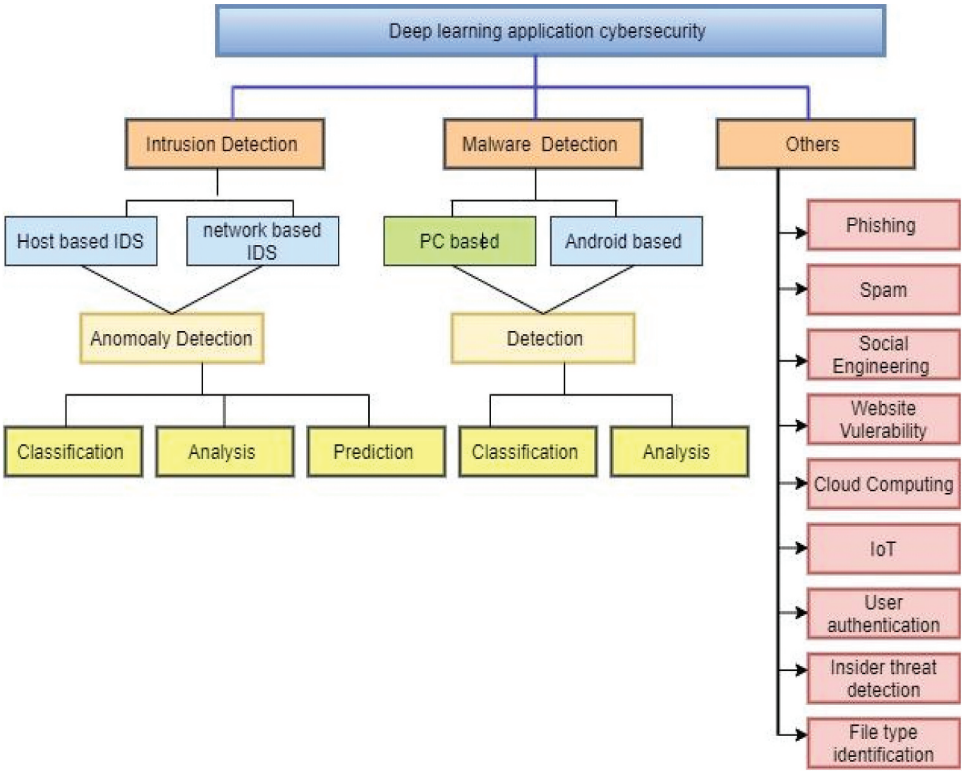



**Figure 13.** Primary areas of using DL in cybersecurity.

of the error are complex to fix. Therefore, in the presentation, the researcher should focus on the dominant elements of the intrusions to develop an effective cybersecurity knowledge technique.

Strong production, CPU, another extensive repository area, and adequate knowledge remain the primary source elements. The DL methods for solving cybersecurity challenges should focus on one specific topic. Instead, the

researcher can consolidate the DL design with various machine learning methods to discover essential data. In addition, the researcher can similarly focus on multiple built-in deep learning models to improve the appearance in the future.

## Conclusion

The rapid technological change makes it a challenging task to secure the systems. Therefore, it is advisable to have a more innovative way to deal with the current situations affecting the taste of deep learning technologies. We show a broad summary of cyber security applications from deep learning approaches. In this study, three DL techniques are examined and discussed. First, the common cyber-attacks are discussed using publicly accessible datasets. Then a proposed framework for cybersecurity is illustrated using DL techniques for general applications. Then, a lab setup is performed to capture live network packets, analyze real-time cyber security attacks, and assess various essential characteristics, namely false alarm rate, detection rate, accuracy, precision, recall, etc. Finally, the researchers' challenges, including technological and operational aspects, are examined, highlighting the future direction of researching DL in cybersecurity.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## ORCID

Sanjay Misra http://orcid.org/0000-0002-3556-9331
Luis Fernandez-Sanz http://orcid.org/0000-0003-0778-0073

## References

Alazab, M., S. Venkatraman, P. Watters, and M. Alazab. 2011. Zero-day malware detection based on supervised learning algorithms of API call signatures. Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1–2 December 2011, Conferences in Research and Practice in Information Technology, Volume 121, Australian Computer Society Inc./ACM pp. 171–81

Amiri, F., M. R. Yousefi, C. Lucas, A. Shakery, and N. Yazdani. 2011. Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications* 34 (4):1184–99. doi:10.1016/j.jnca.2011.01.002.

Banadaki, Y. M. 2020. Detecting malicious dns over https traffic in domain name system using machine learning classifiers. *Journal of Computer Sciences and Applications* 8 (2):46–55. doi:10.12691/jcsa-8-2-2.

Binks, A. 2019. The art of phishing: Past, present and future. *Computer Fraud & Security* 2019 (4):9–11. doi:10.1016/S1361-3723(19)30040-5.

Brown, A., A. Tuor, B. Hutchinson, and N. Nichols, . 2018. "Recurrent neural network attention mechanisms for interpretable system log anomaly detection. *Proceedings of the First Workshop on Machine Learning for Computing Systems*, USA, pp. 1–8 doi:10.1145/3217871.3217872.

Buczak, A. L., and E. Guven. 2015. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communucation Survey Tution* 18 (2):1153–76. doi:10.1109/COMST.2015.2494502.

Chowdhury, M. U., F. Hammond, G. Konowicz, C. Xin, H. Wu, and J. Li. 2017." A few-shot Deep Learning approach for ımproved ıntrusion detection." *Proccedings of the IEEE 8th annual ubiquitous computing*, NY, USA, pp. 456–62 doi:10.1109/UEMCON.2017.8249084.

Costa, K. A. D., J. P. Papa, C. O. Lisboa, R. Munoz, and V. De Albuquerque. 2019. Internet of things: A survey on machine learning-based intrusion detection approaches. *Computer Network* 151:147–57. doi:10.1016/j.comnet.2019.01.023.

Creech, G., and J. Hu. 2013. A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. *IEEE Transactions on Computers* 63 (4):807–19. doi:10.1109/TC.2013.13.

Diro, A. A., and N. Chilamkurti. 2018. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems* 82:761–68. doi:10.1016/j.future.2017.08.043.

Ferrag, M. A., L. Maglaras, S. Moschoyiannis, and H. Janicke. 2020. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. *Journal of Information Security and Applications* 50:102419. doi:10.1016/j.jisa.2019.102419.

Folino, G., and P. Sabatino. 2016. Ensemble based collaborative and distributed intrusion detection systems: A survey. *Journal of Network Computer Application* 66:1–16. doi:10.1016/j.jnca.2016.03.011.

Fontugne, R., P. Borgnat, P. Abry, and K. Fukuda. 2010. "Mawilab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking." *Proceedings of the 6th International Conference, pp* . 1–12. USA. doi:10.1145/1921168.1921179.

Garcia, S., M. Grill, J. Stiborek, and A. Zunino. 2014. An empirical comparison of botnet detection methods. *Computers & Security* 45:100–23. doi:10.1016/j.cose.2014.05.011.

Guo, W., T. Wang, and J. Wei. 2018. "Malware detection with convolutional neural network using hardware events." *Proccedings of the CCF national conference on computer engineering and technology*, pp. 104–15.

Haddadi, F., S. Khanchi, M. Shetabi, and V. Derhami. 2010. "Intrusion detection and attack classification using feed-forward neural network." *2010 Second international conference on computer and network technology*, pp. 262–66.

Hu, J. 2010. Host-based anomaly intrusion detection. In Stavroulakis, P., Stamp, M. edited by *Handbook of information and communication security*, 235–55, Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-04117-4_13.

Jabez, J., and B. Muthukumar. 2015. Intrusion detection system (ids): anomaly detection using outlier detection approach. *Procedia Computer Science* 48:338–46. doi:10.1016/j.procs.2015.04.191.

Jamdagni, A., Z. Tan, X. H. Z, P. Nanda, and R. P. Liu. 2013. Repids: A multi-tier real-time payload-based intrusion detection system. *Computer Networks* 57 (3):811–24. doi:10.1016/j.comnet.2012.10.002.

Jiang, W., H. Li, S. Liu, X. Luo, and R. Lu. 2020. Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles. *IEEE Transspot Vehicles Technology* 69 (4):4439–49. doi:10.1109/TVT.2020.2977378.

Jonker, M., A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti. 2017. "Millions of targets under attack: A macroscopic characterization of the DoS ecosystem." *Proceedings of the 2017 Internet Measurement Conference*, pp. 100–13.

Khan, R. U., X. Zhang, M. Alazab, and R. Kumar. 2019. "An improved convolutional neural network model for intrusion detection in networks." *Cybersecurity and cyberforensics conference*, Melbourne, Australia, pp. 74–77 doi:10.1109/CCC.2019.000-6.

Kim, J., J. Kim, H. L. T. Thu, and H. Kim, . 2016b. "Long short term memory recurrent neural network classifier for intrusion detection." *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, Korea, pp. 1–5 doi:10.1109/PlatCon.2016.7456805.

Kim, G., H. Yi, J. Lee, Y. Paek, and S. Yoon. 2016a. LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems. *arXiv preprint arXiv:1611.01726*.

Koc, L., T. A. Mazzuchi, and S. Sarkani. 2012. A network intrusion detection system based on a hidden naïve bayes multiclass classifier. *Expert Systems with Applications* 39 (18):13492–500. doi:10.1016/j.eswa.2012.07.009.

Koroniotis, N., N. Moustafa, E. S. N, and B. Turnbull. 2019. Towards the development of realistic botnet dataset in the Internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems* 100:779–96. doi:10.1016/j.future.2019.05.041.

Kuang, F., W. Xu, and S. Zhang. 2014. A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing* 18:178–84. doi:10.1016/j.asoc.2014.01.028.

Larson, D. 2016. Distributed denial of service attacks–holding back the flood. *Network Security* 2016 (3):5–7. doi:10.1016/S1353-4858(16)30026-5.

Lippmann, R. P., D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, and M. A. Zissman. 2000. "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation." *Proceedings DARPA Information Survivability Conference and Exposition*. Hilton Head,USA, DISCEX 2:12–26 doi:10.1109/DISCEX.2000.821506.

Loukas, G., E. Karapistoli, E. Panaousis, P. Sarigiannidis, A. Bezemskij, and T. Vuong. 2019. A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles. *Advanced Hoc Network* 84:124–47. doi:10.1016/j.adhoc.2018.10.002.

Low, P. 2017. Insuring against cyber-attacks. *Computer Fraud & Security* 2017 (4):18–20. doi:10.1016/S1361-3723(17)30034-9.

Mahdavifar, S., and A. A. Ghorbani. 2019. Application of deep learning to Cybersecurity: A survey. *Neurocomputing* 347:149–76. doi:10.1016/j.neucom.2019.02.056.

Mahloujifar, S., D. I. Diochnos, and M. Mahmoody. 2019. Learning under p-tampering poisoning attacks. *Annual Mathmateical Artificial Intellegent* 1–34.

Mcintosh, T., J. Jang-Jaccard, P. Watters, and T. Susnjak. 2019. "The inadequacy of entropy-based ransomware detection." *International conference on neural information processing*, Sydney, Australia, pp.181–89 doi:10.1007/978-3-030-36802-9_20.

Milenkoski, A., M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne. 2015. Evaluating computer intrusion detection systems: A survey of common practices. *ACM Computer Survey* 48 (1):12. doi:10.1145/2808691.

Mnih, V., M. AdriaPuigdomenechbadia, A. Mirza, T. Graves, T. Lillicrap, D. Harley, and K. Kavukcuoglu, 2016. "Asynchronous methods for deep reinforcement learning" International *Conference on Machine Learning*, pp. 1928–37.

Mukkamala, S., A. Sung, and A. Abraham. 2006. V. R. Vemuri, *Enhancing Computer Security with Smart Technology.*, CRC Press: USA. 125–63.

Nadiammai, G. V., and M. J. E. I. J. Hemalatha. 2014. Effective approach toward Intrusion Detection System using data mining techniques. *Egyptian Informatics Journal* 15 (1):37–50. doi:10.1016/j.eij.2013.10.003.

Pan, S., T. Morris, and U. Adhikari. 2015. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid* 6 (6):3104–13. doi:10.1109/TSG.2015.2409775.

Radford, A., L. Metz, and Soumithchintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial network. *ArXiv preprint arXiv*:1511.06434.

Rezvy, S., M. Petridis, A. Lasebae, and T. Zebin. 2018, ." Intrusion detection and classification with autoencoded deep neural network." *International Conference on Security for Information Technology and Communications*, Bucharest, Romania, Springer pp. 142–56.

Ring, M., S. Scheuring, D. Landes, A. Hotho, and A. Hotho. 2019. A survey of network-based intrusion detection data sets. *Computers &ampsecurity* 86(*Comput. Secur.*):147–67. doi:10.1016/j.cose.2019.06.005.

Sarker, I. H., S. B. Asm Kayes, H. Alqahtani, P. Watters, A. Ng, and A. Ng. 2020. Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data* 7 (1):1–29. doi:10.1186/s40537-020-00318-5.

Sharafaldin, I., A. H. Lashkari, and A. A. Ghorbani. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* 1:108–16.

Shenfield, A., D. Day, and A. Ayesh. 2018. Intelligent intrusion detection systems using artificial neural networks. *ICT Express* 4 (2):95–99. doi:10.1016/j.icte.2018.04.003.

Shiravi, A., H. Shiravi, M. Tavallaee, and A. A. Ghorbani. 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security* 31 (3):357–74. doi:10.1016/j.cose.2011.12.012.

Shoaib, M., S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga. 2014. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* 14 (6):10146–76. doi:10.3390/s140610146.

Sihag, S., and A. Tajer. 2020. "Secure estimation under causative attacks." IEEE Transactions on Information Theory. doi:10.1109/TIT.2020.2985956

Singh, R., H. Kumar, and R. K. Singla. 2015. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. *Expert Systems with Applications* 42 (22):8609–24. doi:10.1016/j.eswa.2015.07.015.

Tang, T. A., D. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho. 2016. "Deep learning approach for network intrusion detection in software defined networking." *2017 international conference on wireless networks and mobile communications*, pp. 258–63.

Tavallaee, M., E. Bagheri, W. Lu, and A. A. Ghorbani. 2009. "A detailed analysis of the KDD CUP 99 data set." *2009 IEEE symposium on computational intelligence for security and defense applications*, Ottawa, Canada, pp. 1–6 doi:10.1109/CISDA.2009.5356528.

Van, N. T., and T. N. Thinh. 2017. "An anomaly-based network intrusion detection system using deep learning." *2017 international conference on system science and engineering (ICSSE)*, pp. 210–14

Wu, P., and H. Guo. 2019. "LuNET: A deep neural network for network intrusion detection. "*2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China, pp. 617–24 doi:10.1109/SSCI44817.2019.9003126.

Wu, Z., J. Wang, L. Hu, Z. Zhang, and H. Wu. 2020. A network intrusion detection method based on semantic re-encoding and deep learning. *Journal of Network and Computer Applications* 164:102688. doi:10.1016/j.jnca.2020.102688.

Xin, Y., L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang. 2018. Machine learning and deep learning methods for cybersecurity. *IEEE Access* 6:35365–81.

Xiong, C., and Q. Yu. 2018. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. *Applied Energy* 1211:538–48 doi:10.1016/j.apenergy.2017.11.072

Yang, X., K. Lingshuang, L. Zhi, C. Yuling, L. Yanmiao, Z. Hongliang, G. Mingcheng, H. Haixia, and W. Chunhua. 2018a. Machine learning and deep learning methods for cybersecurity. *IEEE Access* 6:35365–81.

Ye, Y., T. Li, D. Adjeroh, and S. S. Iyengar. 2018. A survey on malware detection using data mining techniques. *ACM Computing Surveys (CSUR)* 50 (3):1–40. doi:10.1145/3073559.

Yin, C., Y. Zhu, J. Fei, and X. He. 2017. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access* 5:21954–61. doi:10.1109/ACCESS.2017.2762418.

Zhang, H., X. Yu, P. Ren, C. Luo, and G. Min. 2019. Deep adversarial learning in intrusion detection: A data augmentation enhanced framework. *arXiv preprint arXiv:1901.07949*.

Zhengbing, H., L. Zhitang, and W. Junqi, 2008. "A novel network intrusion detection system (nids) based on signatures search of data mining." *First International Workshop on Knowledge Discovery and Data Mining*, pp. 10–16.