International Journal of IT & Knowledge Management (ISSN: 0973-4414)
Volume 17 • Issue 1 pp. 23-36 June 2023 - Dec 2023 www.csjournals.com

Network Intrusion Detection: A Study Using Machine Learning Techniques

[1]Navroop Kaur, [2]Meenakshi Bansal, [3]Sukhwinder Singh Sran

[1]Research Scholar (Ph.D.), Punjabi University, Patiala, Punjab, India.
[2]Associate Professor, CSE, Yadavindra Department of Engineering, Talwandi Sabo, India.
[3]Assistant professor, Department of Computer Science & Engineering, Punjabi University, Patiala, India.
knavroop7488@gmail.com, ermeenu10@gmail.com, sukhwinder.sran@gmail.com

Abstract:-The rapid expansion of the Internet and communication technologies has led to a massive surge in data transmission. This data has become a prime target for malicious actors who continually develop sophisticated attacks to steal or compromise it. The escalating frequency and complexity of such attacks pose a significant threat to system security, making intrusion detection one of the most critical challenges in cyber security. Intrusion Detection Systems (IDS) are essential tools designed to monitor and analyze network traffic for signs of malicious activity. Despite extensive research and development in this field, existing IDS solutions often struggle with achieving high detection accuracy while maintaining low false alarm rates. Moreover, detecting zero-day attacks remains a persistent challenge.

Keywords:-IDS, Machine learning and network security.

1. Introduction: - The exponential rise in network traffic due to the proliferation of IoT devices, cloud computing, and mobile technologies has made traditional security mechanisms inadequate. Network Intrusion Detection Systems (NIDS) serve as a crucial line of defense by monitoring and analyzing network traffic to detect malicious activities. However, rule-based systems often fail to detect novel or sophisticated attacks. Machine Learning (ML) offers a promising solution by enabling systems to learn from data and identify patterns associated with intrusions. This paper studies the effectiveness of various ML models in detecting network intrusions [1,8].

In the 1990's, a new method of detection was explored to face the increasing number of attacks. This method was anomaly detection and it looked for unusual behaviour/activity in a system to raise an alarm. Nevertheless, the inconsistent nature of the networks, between the 1990s and the 2000s, caused a lot of false alarms. Thus, many administrators stopped using IDS due to their unreliability.

An Intrusion Detection System (IDS) is a security tool either hardware- or software-based that monitors network activity and raises alerts when malicious behavior is detected. Often referred to as the "watchful eye" of the network, an IDS plays a critical role in the security infrastructure of modern digital environments. It enables the early detection of cyber-attacks, providing administrators with a valuable window of opportunity to respond and mitigate potential threats.

IDSs are capable of identifying a wide range of attack types, such as Denial of Service (DoS) and Man-in-the-Middle (MitM) attacks. They can monitor and log all specified network traffic, offering real-time visibility into suspicious activities. By providing detailed information about attacks as they happen, IDSs assist security professionals in analyzing threat patterns and preparing defenses against similar future incidents.

Model	Type	Key Features	Pros	Cons
Decision Tree (DT)	Supervised	Hierarchical rule-based classification	Easy to interpret, fast	Prone to over fitting
Random Forest (RF)	Supervised	Ensemble of decision trees	High accuracy, handles missing data	Slower than single tree
Support Vector Machine (SVM)	Supervised	Hyper plane-based separation	Effective in high-dimensional spaces	High computational cost
K-Nearest Neighbors (KNN)	Supervised	Distance-based classification	Simple to implement	Performance drops with large datasets
Naive Bayes (NB)	Supervised	Probabilistic classifier	Fast, works well with small datasets	Assumes feature independence
Deep Neural Networks	Supervised	Multi-layered nonlinear architecture	High detection accuracy,	Requires large datasets, less

(DNN)			adaptable	interpretable
Auto encoders	Unsupervised	Dimensionality reduction + anomaly detection	Effective for unknown attacks	Needs tuning, less interpretable

Table 1: Machine Learning Models for Intrusion Detection

There are two main categories of IDS:-

1.1 Network Intrusion Detection Systems (NIDS), which monitor traffic across an entire network.

1.2 Host Intrusion Detection Systems (HIDS), which focus on monitoring activity on individual devices or hosts.

1.1 A Network Intrusion Detection System (NIDS) **is** a security tool designed to monitor and analyze network traffic in real-time to detect unauthorized access, malicious activities**,** or policy violations within a computer network. It acts as a passive monitoring system, alerting administrators when suspicious patterns, known attack signatures, or anomalies are detected [3,7].

Function	Explanation
Traffic Monitoring	Continuously inspects incoming and outgoing packets across a network segment.
Attack Signature Matching	Compares traffic against a database of known attack patterns.
Anomaly Detection	Identifies deviations from normal behavior that may indicate unknown threats.
Alert Generation	Notifies security teams about potential intrusions or suspicious activity.
Logging & Reporting	Keeps records of detected events for forensic analysis and auditing.

Table 2:- Core Functions of a NIDS

Technique	Description
Signature-based	Detects attacks by comparing traffic to known patterns (e.g., Snort rules).
Anomaly-based	Learns normal behavior and flags deviations (e.g., using machine learning).
Hybrid	Combines both techniques for improved detection coverage and accuracy.

Table 3: - Types of NIDS Detection Techniques

Advantages

- Early warning for network attacks
- Detects both external intrusions and internal misuse
- Helps enforce organizational security policies

1.2 Host Intrusion Detection Systems (HIDS)

Host Intrusion Detection Systems (HIDS) focus on monitoring and analyzing activities on individual devices or hosts (such as computers or servers) to detect suspicious behavior. Unlike NIDS, which analyze network traffic, HIDS operate at the system level and can detect unauthorized access, file modifications, or abnormal system calls [10].

Aspect	Host Intrusion Detection Systems (HIDS)
Monitoring Focus	Individual host systems (files, processes, logs)
Deployment	Installed on each host (workstation or server)
Detection Capabilities	Detects insider threats, malware, privilege escalation
Advantages	Detailed visibility, accurate logging, effective for encrypted traffic
Disadvantages	High resource usage, harder to manage at scale, may miss network-level attacks

Table 4: HIDS deployment with Advantages and Disadvantages

Advantages of HIDS:

- Can detect unauthorized changes to system files
- Suitable for encrypted traffic where NIDS may fail
- Works well in combination with antivirus or endpoint tools

Disadvantages of HIDS:

- Requires installation on each host, increasing maintenance effort
- Limited view of external network activity
- May generate false positives from legitimate user behavior

2. **Machine Learning Concepts**: - Machine Learning (ML) is a core component of Artificial Intelligence (AI) that involves training algorithms to identify patterns within data. Through this training process, a predictive model is developed, enabling the system to make informed decisions or automate tasks. In the context of Intrusion Detection Systems (IDS), machine learning can be effectively used to identify both known and previously unseen (unknown) attacks, provided the model is trained on sufficiently representative and diverse datasets.

2.1 Supervised Machine Learning

Supervised machine learning involves training a model using labeled data to learn a function that maps inputs to corresponding outputs. This approach relies on a dataset where both the input features and the desired output labels are known. Based on this data, the model learns patterns that allow it to make predictions on new, unseen data [8].

Supervised learning is broadly categorized into two types: **classification** and **regression**.

- **Classification Models:-**Classification models are used to assign input data into specific predefined categories based on its features. During training, the model learns from labeled input-output pairs to identify distinguishing characteristics of each class. These models are particularly effective in intrusion detection systems (IDS), where they can distinguish between normal and malicious network traffic. For example, a well-trained classification

model can identify incoming traffic as either legitimate or abnormal. Furthermore, it can categorize abnormal traffic into known attack types such as Denial of Service (DoS), phishing, worms, or port scans. Common classification algorithms include Decision Trees, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forests, and Neural Networks[6].

- **Regression Models:**-Regression models are designed to predict continuous numerical values by learning the relationship between independent variables (inputs) and a dependent variable (output). These models are typically used for forecasting and trend analysis for instance, predicting stock prices or monitoring performance metrics over time [5]. In the context of IDS, regression can be applied in performance evaluation or to forecast network behavior under certain conditions. Common regression techniques include Linear Regression, Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines.

2.2 Unsupervised Machine Learning

Unsupervised machine learning is applied to datasets without labeled outputs. As the name implies, it operates without direct supervision or guidance from the user. Instead, the model autonomously learns from the data by identifying hidden patterns, structures, and relationships. The algorithm typically organizes the data into groups or clusters based on similarities or differences among the features. Unsupervised learning is particularly effective in exploring and analyzing large datasets, making it a powerful tool in big data analytics[9,10]. It is commonly applied in three major problem types: clustering**,** association, and dimensionality reduction**.**

Aspect	Supervised Learning	Unsupervised Learning
Definition	Learns from labeled data	Learns from unlabeled data
Purpose	Predict outcomes, classify or regress	Discover patterns or groupings
Algorithms	Decision Tree, SVM, Random Forest, DNN	K-Means, Hierarchical Clustering, Auto encoders, PCA
Dataset Example	NSL-KDD (labeled attack types: DoS, Probe, etc.)	CIC-IDS2017 (used for discovering unknown patterns)

Aspect	Supervised Learning	Unsupervised Learning
Output	Predictive labels (e.g., Normal or Attack)	Clusters, anomalies, or representations
Advantages	High accuracy, strong generalization for known patterns	Good for anomaly or novel pattern detection
Disadvantages	Requires a lot of labeled data, struggles with unseen data	May produce false positives, hard to evaluate objectively
Application in IDS	Detecting known intrusions	Discovering unknown or zero-day intrusions

Table 5:- Supervised learning V/S Unsupervised learning

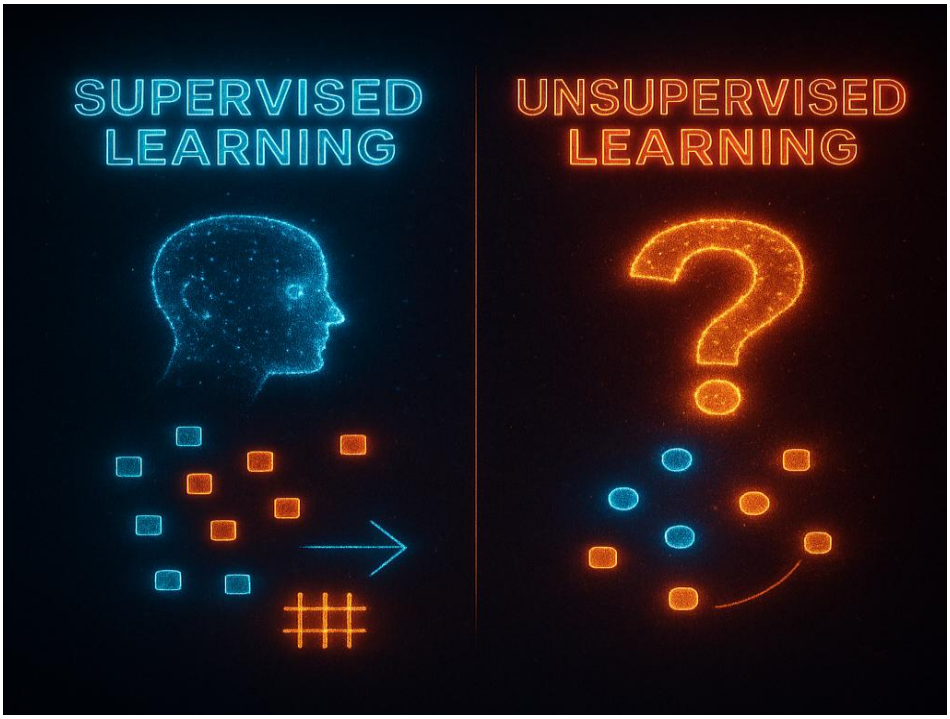

Image 1: Supervised V/s Unsupervised learning concepts

Algorithm	Type	Accuracy	Training Time	Strengths	Weaknesses
Decision Tree (DT)	Supervised	★★★★☆	Fast	Interpretable, handles both types of features	Prone to overfitting
Random Forest (RF)	Ensemble	★★★★★	Moderate	Robust, high accuracy, handles imbalance	Slower, less interpretable
SVM	Supervised	★★★★☆	Slow (large data)	Good with high-dimensional data	Poor scalability, sensitive to kernel
Naïve Bayes (NB)	Supervised	★★★☆☆	Very fast	Simple, fast, good baseline	Assumes feature independence
k-NN	Supervised	★★★☆☆	Slow (at test)	Simple, non-parametric	Computationally expensive
Logistic Regression	Supervised	★★★★☆	Fast	Interpretable, probabilistic output	Assumes linear boundaries
K-Means Clustering	Unsupervised	★★☆☆☆	Fast	Easy to implement, good for anomaly detection	Poor cluster quality with noise
Autoencoder (AE)	Unsupervised DL	★★★★☆	Moderate–Slow	Learns latent patterns for anomaly detection	Needs careful architecture tuning
ANN / DNN	Supervised DL	★★★★★	Slow	Detects complex, nonlinear patterns	Needs large data and compute
Gradient Boosting (XGBoost, LightGBM)	Ensemble	★★★★★	Moderate–Fast	Highly accurate, handles missing data	Slightly harder to tune

Table 6:- Machine Learning Algorithms: Performance Analysis for IDS

3. Data sets: - To train and test their models used datasets. The most known and used datasets for IDS testing:-

3.1 KDD cup99

The **KDDCup99** dataset is one of the most extensively used benchmarks for evaluating Intrusion Detection Systems (IDS). It is derived from the DARPA'98 dataset and contains approximately

4.9 million samples, each comprising 41 features[11]. Every sample is labeled as either **Normal** or an **Attack**, with attack instances further categorized into four types: Denial of Service (DoS)**,** User to Root (U2R)**,** Remote to Local (R2L)**, and** Probe. The dataset is available in three variants:

1. The **full dataset**,
2. A **10% subset** of the full data,
3. A **test dataset** containing **311,029 samples**.

Despite its popularity, the KDDCup99 dataset has notable limitations. One major issue is **class imbalance** dominant classes such as DoS and Probe have a high number of similar samples, while rare classes like R2L and U2R are underrepresented.[7] Depending on the subset used, some classes may even be entirely missing, which can significantly affect the training and evaluation of IDS models.

3.2 CICIDS 2017

The **CICIDS2017** dataset was developed in 2017 by the Canadian Institute for Cyber security (CIC) to provide a realistic benchmark for evaluating intrusion detection systems. It was generated using real-world network traffic that includes both normal behaviour and recent**,** diverse cyber-attacks[6]. The dataset was analysed using CICFlowMete**r**, which captured details such as timestamps**,** source and destination IP addresses**,** protocols**, and** attack types. It includes a wide range of commonly encountered attacks, such as Brute Force FTP**,** Brute Force SSH**,** Denial of Service (DoS)**,** HeartBleed**,** Web Attacks**,** Infiltration**,** Botnet**, and** Distributed Denial of Service (DDoS)**.**

This study highlights the significant role of machine learning in enhancing the efficiency of Intrusion Detection Systems (IDSs). A key factor influencing IDS performance is the quality of the dataset used during training. Many of the reviewed research works utilize labeled datasets to effectively train machine learning models. However, as datasets continue to grow in size, traditional machine learning techniques often struggle with scalability and performance.

To address this, researchers are increasingly turning to deep learning models such as Convolutional Neural Networks (CNNs) which can automatically extract meaningful features from raw data. These models have shown strong potential in improving Network Intrusion Detection Systems (NIDS), especially for detecting zero-day attacks. Nevertheless, these advanced techniques come with trade-offs: they demand more computational power, longer training times, and frequent updates with real-world network traffic to maintain effectiveness[12].

Feature	KDD Cup 99	CICIDS 2017
Year of Release	1999	2017
Data Type	Simulated, outdated network traffic	Realistic, modern network traffic
Protocols	TCP, UDP, ICMP	HTTP, HTTPS, FTP, SSH, DNS, etc.
Total Features	41	78
Attack Types	4 main types (DoS, U2R, R2L, Probe)	14 types (Brute Force, Botnet, etc.)
Label Distribution	Highly imbalanced	More balanced across classes
Data Size	~5 million records	~3 million records (CSV format)
Realism	Synthetic attacks	Real-world traffic + attacks
Usage	Benchmark dataset	Research-grade & industry relevant

Table 7:- Comparison between KDD cup99 and CICIDS 2017 (Datasets)

The **KDD Cup 99** and **CICIDS 2017** datasets using basic feature statistics, class distributions, and shape comparison. This assumes you have the datasets in CSV format (kdd.csv and cicids.csv):-

```
import pandas as pd

# Load datasets

kdd_df = pd.read_csv('kdd.csv')

cicids_df = pd.read_csv('cicids.csv')

# Print basic shapes

print("KDD Cup 99 shape:", kdd_df.shape)
```

```
print("CICIDS 2017 shape:", cicids_df.shape)

# Feature counts

print("KDD Cup 99 features:", len(kdd_df.columns))

print("CICIDS 2017 features:", len(cicids_df.columns))

# Unique classes (assuming 'label' is the target column)

print("KDD Classes:", kdd_df['label'].nunique())

print("CICIDS Classes:", cicids_df['label'].nunique())

# Sample distribution by class

print("\nKDD Class Distribution:")

print(kdd_df['label'].value_counts())

print("\nCICIDS Class Distribution:")

print(cicids_df['label'].value_counts())
```

Algorithm	KDD99 Accuracy	CICIDS2017 Accuracy
Decision Tree	~98%	~92–95%
Random Forest	~99%	~95–97%
SVM	~97%	~92–94%
Logistic Regression	~96%	~90–93%
ANN / DNN	~98–99%	~96–98%

Table 8 :-Dataset Performance Snapshot (Typical from KDD99 & CICIDS2017)

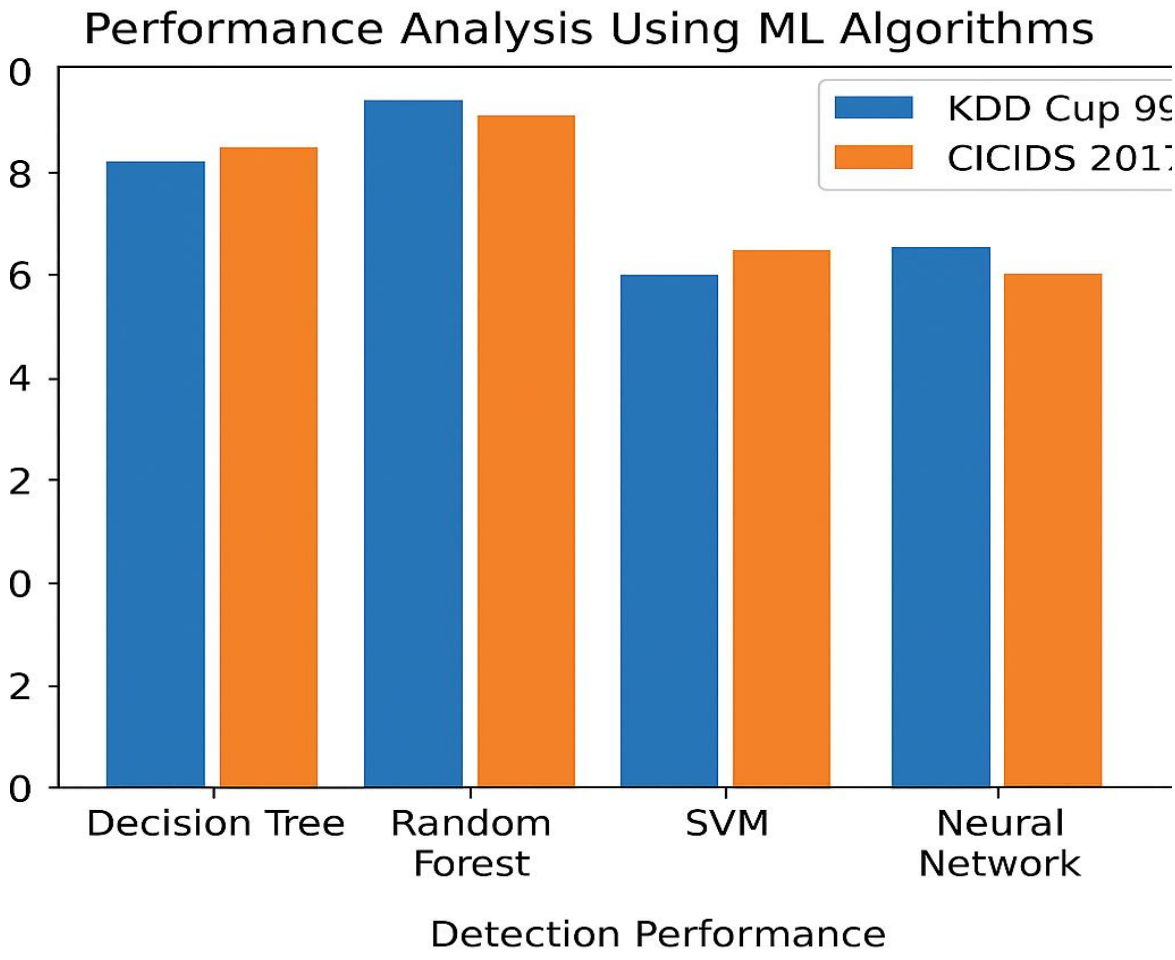

Image 2: Performance analysis using ML Algorithms

4. Conclusion:-Datasets play a critical role in the development and evaluation of Intrusion Detection Systems (IDSs). Analysis shows that KDDCup99 and NSL-KDD are still used in approx. **56%** of IDS testing studies. Although these datasets are well-established and widely referenced, they are significantly outdated. Since their creation, network architectures have evolved dramatically, with the widespread adoption of IoT and wireless technologies leading to a massive increase in data traffic and the emergence of sophisticated new cyber threats.

As a result, IDS models trained solely on these older datasets are unlikely to perform effectively in modern, real-world environments. This underscores the importance of using recent,

representative datasets to train and evaluate IDS solutions, ensuring they remain accurate, adaptive, and robust against today's dynamic cyber threats.

5. References:-

1. Seldon. Machine Learning Regression Explained, https://www.seldon.io/machine-learning-regression-explained (accessed on 19 May 2022).

2. IBM Cloud Education. Machine Learning,https://www.ibm.com/cloud/learn/machine-learning (accessed on 19 May 2022).

3. Protic, D. Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ Datasets. Vojnoteh. Glas. 2018, 66, 580–596

4. Wisanwanichthan, T.; Thammawichai, M. A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM. IEEE Access 2021, 9, 138432–138450.

5. Buczak, A. L., & Guven, E. (2016).A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.[https://doi.org/10.1109/COMST.2015.2494502].

6. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58. [https://doi.org/10.1145/1541880.1541882].

7.Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018).A deep learning approach to network intrusion detection.IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50. [https://doi.org/10.1109/TETCI.2017.2772792].

8.Xia, Y., Wang, X., Zhang, X., & Sun, L. (2018).A fault-tolerant ensemble classification framework using noise elimination for intrusion detection. Computers & Security, 73, 198–211. [https://doi.org/10.1016/j.cose.2017.11.011].

9.Yin, C., Zhu, Y., Fei, J., & He, X. (2017).A deep learning approach for intrusion detection using recurrent neural networks.IEEE Access, 5, 21954–21961.[https://doi.org/10.1109/ACCESS.2017.2762418].

10. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009).A detailed analysis of the KDD CUP 99 data set. Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications.[https://doi.org/10.1109/CISDA.2009.5356528].

11. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP 2018: Proceedings of the 4th International Conference on Information Systems Security and Privacy.

12. Aburomman, A. A., & Reaz, M. B. I. (2016).A novel SVM–KNN–PSO ensemble method for intrusion detection system. Applied Soft Computing, 38, 360–372. [https://doi.org/10.1016/j.asoc.2015.10.025].

13.Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset. Information Security Journal: A Global Perspective, 25(1-3), 18–31. [https://doi.org/10.1080/19393555.2015.1125974].