DeepShield: A Hybrid Deep Learning Approach for Effective Network Intrusion Detection

Hongjie Lin*
School of Economics and Management, Xiamen University of Technology, Xiamen, 361024, China

***Abstract*—In today's rapidly evolving digital landscape, ensuring the security of networks and systems has become more crucial than ever before. The ever-present threat of hackers and intruders attempting to disrupt networks and compromise online services highlights the pressing need for robust security measures. With the continuous advancement of security systems, new dangers arise, but so do innovative solutions. One such solution is the implementation of Network Intrusion Detection Systems (NIDSs), which play a pivotal role in identifying potential threats to computer systems by categorizing network traffic. However, the effectiveness of an intrusion detection system lies in its ability to prepare network data and identify critical attributes necessary for constructing robust classifiers. In light of this, this paper proposes, DeepShield, a cutting-edge NIDS that harnesses the power of deep learning and leverages a hybrid feature selection approach for optimal performance. DeepShield consists of three essential steps: hybrid feature selection, rule assessment, and detection. By combining the strengths of machine learning and deep learning technologies, a new solution is developed that excels in detecting network intrusions. The process begins by capturing packets from the network, which are then carefully preprocessed to reduce their size while retaining essential information. These refined data packets are then fed into a deep learning algorithm, which employs machine learning characteristics to learn and test potential intrusion patterns. Simulation results demonstrate the superiority of DeepShield over previous approaches. NIDS achieves an exceptional level of accuracy in detecting malicious attacks, as evidenced by its outstanding performance on the widely recognized CSE-CIC-DS2018 dataset.**

Keywords—Network intrusion detection system; IDS; cyber security; machine learning; deep learning

I. Introduction

The Internet has evolved into a necessary tool and one of the most reliable sources of knowledge about the modern world. It can be considered a crucial component of education and business. Therefore, preserving data across the Internet becomes challenging [1]. Nowadays, internet security is a serious problem [2]. Over the last decade, computer networks have grown in complexity, usage, and size. Cloud computing and the Internet of Things (IoT) have evolved into entirely new types of devices and networks [3]. These networks and systems have grown in size and complexity, so their security has become a critical concern [4]. According to CyberEdge group statistics, the number of attacks on large enterprise networks worldwide has increased significantly in recent years [5]. Advanced Persistent Threats (APT), malware, and denial of service attacks are examples of these attacks [6]. APTs are particularly hazardous and expensive since they are long-term, targeted operations carried out by sophisticated perpetrators targeting the public sector and business enterprises in order to exfiltrate data and cause infrastructure damage [7]. According to cybersecurity studies, these attacks were active for an average of 184 days in 2018 (the duration of attack effectiveness before it is detected) [8].

As a primary layer of defense against computer system vulnerabilities and attacks, a robust security model implements industry-standard security standards such as authorization, access control, confidentiality, and other security requirements [9]. Nevertheless, attacks are likely to continue to present a threat due to vulnerabilities in the system, operational errors, and other issues [10]. Intrusion Detection Systems (IDSs) are critical in identifying and alerting system administrators to intrusions into computers and networks [11]. The IDS can be installed on individual servers within a network, at a centralized location, or distributed around the network [12]. A Network Intrusion Detection System (NIDS) is a kind of IDS intended to track attacks across multiple hosts instead of a single host. These systems monitor network operations using network telemetries, such as network traffic, network flow metadata, and host event logs, to identify attack events [13].

In the realm of NIDS, the convergence of machine learning, artificial intelligence, meta-heuristic algorithms, deep learning, feature selection, association rule mining, and fault diagnosis plays a pivotal role in fortifying cybersecurity defenses and safeguarding critical network infrastructures against evolving cyber threats. Machine learning and artificial intelligence techniques enable IDSs to continually learn and adapt to new attack patterns, enhancing their accuracy in distinguishing between normal and malicious network activities [14-17]. The integration of meta-heuristic algorithms optimizes the performance of intrusion detection models, fine-tuning parameters and reducing false positives [18]. Deep learning empowers systems to automatically extract intricate features from raw network data, enabling the identification of sophisticated and novel attack signatures [19, 20]. Feature selection techniques help to identify the most relevant network attributes, streamlining the detection process and reducing computational overhead [21]. Additionally, fault diagnosis capabilities enable swift identification and response to potential anomalies, further strengthening the overall resilience of intrusion detection systems [16, 22]. Association rule mining holds paramount importance in NIDS as it enables the discovery of hidden patterns and correlations in network data, facilitating the identification of suspicious and anomalous activities that might go undetected using traditional IDSs [23]. This amalgamation of cutting-edge technologies empowers organizations to proactively detect, mitigate, and thwart cyber threats, ensuring the confidentiality, integrity, and availability of critical data and establishing robust and future-proofed NIDs.

To identify intrusions and anomalies, a NIDS continuously monitors network traffic. In the case of high network throughput, using a single NIDS on a network can cause congestion. Deep packet inspection may include significant similarities to complicated signatures of attack rules [24, 25]. Pattern matching is a time-consuming procedure that requires substantially more computing power than a firewall, which might cause a NIDS to become overloaded [26]. When a NIDS becomes overburdened and begins dropping or ignoring packet content, network security may be compromised. Finally, some vulnerabilities may remain unnoticed since some packets associated with the same attack may escape the NIDS's inspection, leading to an insufficient match between packets [27]. NIDS employ several strategies to handle high levels of network traffic, including:

- Hardware upgrades, including the addition of dedicated packet capture cards and more computing resources, as well as modifying the NIDS software to increase its capacity.
- Utilizing a cluster of NIDSs and distributing signature rules and network traffic among the NIDS hosts.

The first strategy, which involves optimizing the NIDS application and upgrading hardware, is prohibitively expensive and unscalable. Every four years, network bandwidth rates grow by a factor of 10; therefore, maintaining a NIDS requires ongoing hardware upgrades. Adjusting a NIDS to cope with greater traffic is a difficult procedure that includes various trade-offs, resulting in the NIDS being more complex than anticipated. The second strategy, which relies on NIDS clusters, is cost-effective and scalable. When network traffic is low, the solution can be adjusted to accommodate it, and resources can be released and utilized for other reasons. Numerous studies have demonstrated the benefit of low-cost clustering computers equipped with NIDS to manage high network traffic loads. Additionally, the cluster can be expanded by adding additional NIDS instances. However, both the distribution of traffic among NIDS instances and the distribution of signature rules are critical to the effectiveness of the solution [28].

The use of machine learning algorithms in the context of NIDS has received considerable attention. Training machine learning algorithms on normal and attack traffic enables them to detect novel differences in network traffic. Traditionally, the NIDS is designed by an expert human analyst who codifies rules defining normal behavior and intrusions [29]. Due to the numerous failures of this method to identify novel intrusions and the aim to reduce the analyst's effort, machine learning algorithms have been incorporated into NIDS to automate the process and supplement the human effort. This study proposes a new method comprised of machine learning and deep learning algorithms for feature selection and intrusion detection.

The key contributions of this research paper include the development of a cutting-edge NIDS that leverages deep learning and a hybrid feature selection approach. The three-step architecture, consisting of hybrid feature selection, rule assessment, and detection, enhances the effectiveness of the intrusion detection system. By combining the strengths of machine learning and deep learning technologies, the proposed NIDS demonstrates superior performance in detecting network intrusions. The careful preprocessing of network data, followed by the application of a deep learning algorithm, allows for the identification and testing of potential intrusion patterns. Simulation results showcase the exceptional accuracy of the proposed method, surpassing previous approaches, as demonstrated by its outstanding performance on the widely recognized CSE-CIC-DS2018 dataset.

II. Importance of the NIDS

As the Internet is vulnerable to various threats, it is critical to develop a system that protects the data and the individuals using it [30]. For years, the scientific community has focused on identifying cyber-attacks that target information and communication networks. Developing a comprehensive and efficient NIDS is one of the primary challenges in network security. These systems are critical for network administrators to detect different security vulnerabilities within an organization's network. The NIDS monitors and analyses network traffic incoming and departing an organization's network devices and triggers alerts if an intrusion is detected.

An IDS takes its name from the conjunction of two concepts, intrusion and detection systems. Generally, an intrusion is defined as gaining access without authorization to a network or computer system with the intent of compromising its functionality, privacy, or reliability. The IDS detects such illegal activities. Therefore, the IDS serves as a security component responsible for monitoring network traffic to identify suspicious activities that violate security policies and endanger the network's availability, reliability, and stability. It notifies hosts or network administrators of detected malicious activities. As shown in Fig. 1, NIDS is deployed passively by connecting to a network switch equipped with mirror ports. In order to monitor traffic and detect intrusions, all inbound and outbound network traffic should be mirrored to NIDS. By installing NIDS in the middle of the network switch and firewall, all traffic can be routed through it.

Modern NIDS are divided into two categories: rule-based misuse detection and statistical anomaly detection. In the first method, a database is used to store the characteristics of a wide variety of known attacks and the network traffic is classified as an "attack" if the retrieved characteristics match those stored in the database. Although this kind of NIDS can rapidly and accurately detect known attacks, it is weak at detecting future attacks. As a result, anomaly detection-based NIDS has gained popularity recently. According to its basic assumptions, the system detects and identifies abnormalities in network traffic properties or distributions [31].

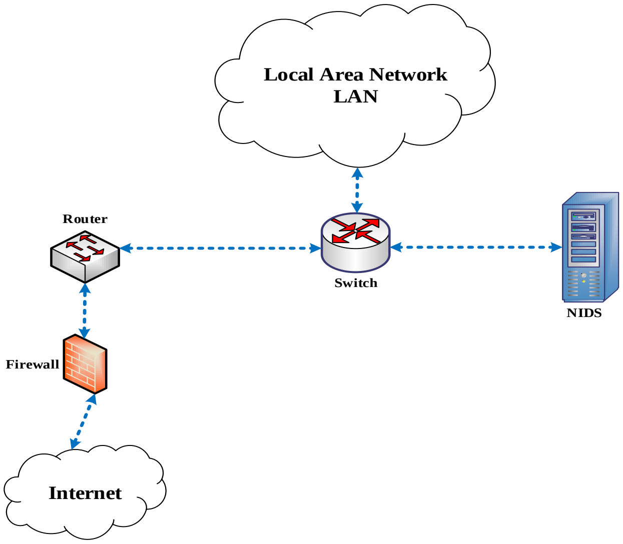

Fig. 1. Passive deployment of NIDS.

A wide range of machine learning algorithms have been implemented in NIDS to detect anomalies. Different machine learning algorithms have been used to discriminate normal from abnormal network activity, including Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT). Nevertheless, as attack categories diversify and network traffic grows, shallow learning approaches cannot be applied effectively to large-scale NIDS. Recently, deep learning has been the subject of extensive research owing to its ability to generate features automatically. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Multi-Layer Perceptron (MLP) have also been incorporated into NIDS.

According to [32], When dealing with large datasets, deep learning-based NIDS perform better. Nevertheless, these NIDS approaches are limited in certain respects. First, the majority of them fail to indicate the nature of the attack in their categorization results. Since various attacks demand distinct defense mechanisms in actual systems, detecting "normal" or "abnormal" is inadequate. Second, most of these approaches are evaluated using NSL KDD, or KDD99 datasets gathered around 20 years ago. New attacks emerge practically daily; thus, relying on historical traffic statistics does not accurately represent the effectiveness of NIDS in modern networks. Experiments are conducted on a subset of the dataset without considering the system's performance as a whole. Lastly, they ignore the effects of class imbalances on classification performance, which results in a large reduction in detection rates, particularly for minority classes.

III. Related Work

Generally, NIDSs are categorized into three main classes: anomaly-based, misuse-based, and hybrid. Misuse-based techniques detect intrusions using a pattern-matching model. Due to the fixed model, this approach is capable of detecting the most common types of attacks with a high degree of accuracy. Nevertheless, this characteristic also presents an inherent disadvantage since a dynamic environment may give rise to novel attacks or variations at any time. A second kind of IDS strategy is anomaly-based, which relies solely on normal data in order to identify abnormal samples. The NIDS raises alarms in the event of a real-world attack using the misuse-based technique, but it does not offer any additional information about the type of attack. The drawback of this technique is that it performs poorly in terms of accuracy because some attacks resemble normal data, or the extracted properties are difficult to distinguish between attack and normal data.

Over the last decade, machine learning algorithms have gained much attention from researchers in developing IDSs. Anwer, et al. [33] have proposed a method for selecting features that consider irrelevant and redundant features. The method applies different strategies based on the selection of filter and wrapper features. It achieves a high level of accuracy by selecting the minimum number of features. Experimental results are presented using the UNSW-NB15 dataset. Tian, et al. [34] have suggested a robust and sparse technique based on a one-class support vector machine (OSVM) to find samples that vary from the majority of data. The Ramp loss function has been used to enhance the performance of this model, making the approach more robust and sparser.

The NSL-KDD dataset is used in the study presented in [35]. The dataset in this research is normalized and discretized using the k-means technique. Feature selection is made using the Information gain algorithm and then submitted to the Naive Bayes machine learning algorithm. They discovered that the k-means clustering approach outperforms the mean and standard deviation discretization methodology. The data is sent to the information-gain technique after it has been labeled using the k-means approach, which employs scoring methods for nominal or weighting continuous qualities that are discredited by applying the maximum entropy. The k-means technique cannot handle nonlinear or incomplete data, one of its key shortcomings. The system's accuracy and false-positive rate may be enhanced further.

Kan, et al. [36] have introduced a novel approach for intrusion detection in IoT networks called Adaptive Particle Swarm Optimization Convolutional Neural Network (APSO-CNN). The approach utilizes the PSO algorithm with a change of inertia weight to dynamically optimize the structure parameters of a one-dimensional CNN. To achieve this, the cross-entropy loss function value of the validation set, obtained from the initial training of the CNN, is utilized as the fitness value for PSO. This adaptive optimization process ensures efficient parameter tuning for improved performance. A new evaluation method is defined that considers both the prediction probability assigned to each category and the prediction label. This evaluation method enables a comprehensive comparison between the proposed APSO-CNN algorithm and manually set parameters for CNN (R-CNN). Furthermore, a comparison is conducted between the proposed APSO-CNN and three other well-known algorithms using five traditional evaluation indicators and accuracy statistical characteristics from ten independent experiments. The simulation results reveal that the APSO-CNN algorithm proves to be effective and reliable for multi-type IoT network intrusion attack detection tasks.

Andresini, et al. [37] have proposed an innovative intrusion detection method that focuses on analyzing the flow-based characteristics of network traffic data. Their approach leverages deep metric learning, which combines autoencoders and Triplet networks to create an effective intrusion detection model. During the training stage, two separate autoencoders are trained using historical normal network flows and attack data, respectively. The autoencoders are designed to reconstruct the original network flow data. Subsequently, a Triplet network is trained to learn an embedding of the feature vector representation of the network flows. This embedding ensures that each flow is positioned close to its reconstruction by the autoencoder associated with the same class (normal or attack) and far away from its reconstruction by the autoencoder of the opposite class. In the predictive stage, when presented with a new network flow, the method assigns it to the class associated with the autoencoder that provides the closest reconstruction of the flow in the embedding space. This process capitalizes on the learned embedding from the training stage and effectively detects potential signs of malicious activities in the network traffic. The results of their proposed methodology demonstrate superior predictive accuracy compared to competitive intrusion detection architectures when evaluated on benchmark datasets. The combination of deep metric learning, autoencoders, and Triplet networks empowers their intrusion detection approach to achieve impressive performance in detecting new instances of malicious behavior within network traffic.

Ravi, et al. [38] have presented an end-to-end model for network attack detection and classification, leveraging deep learning-based recurrent models. Their proposed approach involves extracting features from the hidden layers of recurrent models and utilizing a kernel-based principal component analysis (KPCA) feature selection method to identify optimal features. These optimal features from recurrent models are then combined and used for classification through an ensemble meta-classifier. Extensive experimental analysis and evaluation of the proposed method were conducted on multiple benchmark network intrusion datasets. The results demonstrated that the proposed approach outperformed existing methods as well as commonly used machine learning and deep learning models. In particular, the proposed method achieved a remarkable maximum accuracy of 99% for network attack detection and 97% for network attack classification when applied to the SDN-IoT dataset. Similarly impressive performances were obtained on other network intrusion datasets, including KDD-Cup-1999, UNSW-NB15, WSN-DS, and CICIDS-2017.

Talukder, et al. [39] have introduced a novel hybrid model that combines machine learning and deep learning techniques to achieve higher detection rates while ensuring dependable results. The proposed method focuses on efficient pre-processing by utilizing SMOTE for data balancing and XGBoost for feature selection. To evaluate the effectiveness of their developed method, they conducted a comparison with various machine learning and deep learning algorithms. The goal was to identify the most efficient algorithm to incorporate into the detection pipeline. Through benchmarked performance analysis criteria, they selected the most effective model for network intrusion detection. Their method was tested on two datasets, KDDCUP'99 and CIC-MalMem-2022, and produced remarkable results. The accuracy achieved was 99.99% for KDDCUP'99 and 100% for CIC-MalMem-2022, showcasing the superior performance of the proposed hybrid model. Additionally, their method exhibited no signs of overfitting or issues related to Type-1 and Type-2 errors.

Mohamed and Ejbali [40] have introduced a novel deep reinforcement learning model that effectively combines a SARSA-based reinforcement learning algorithm with a deep neural network for intrusion detection systems. The primary objective of their proposed deep SARSA model is to enhance the detection accuracy of modern and complex attacks in network environments. To validate the performance of their method, they conducted experiments using two well-known benchmark datasets, NSL-KDD and UNSW-NB15. By comparing their approach with various classic machine learning and deep learning models, as well as other published results in the field, they demonstrated that their proposed approach outperforms the other models across multiple evaluation metrics, including accuracy, recall, precision, and F1-score. The integration of deep reinforcement learning, SARSA-based algorithm, and deep neural networks has proven to be a successful strategy for achieving superior intrusion detection accuracy. The proposed approach addresses the challenges posed by modern and complex attacks in network environments, making it a valuable contribution to the field of network security.

IV. Problem Statement

The rapid pace of technological advancements in network and hardware devices presents significant challenges for the implementation and enhancement of intrusion detection systems (IDSs). To fully understand these challenges, it is important to delve into their specific details and implications. Firstly, the challenge of diversity arises from the continuous development of network protocols. As these protocols evolve, it becomes increasingly difficult to differentiate between normal and abnormal data traffic. This poses a significant hurdle for IDSs, as they need to accurately identify potential threats amidst a wide range of network activities. Another challenge is related to low-frequency attacks. The distribution of attack types is often imbalanced, with some occurring less frequently than others. This imbalance negatively impacts the detection precision of IDSs, particularly those utilizing data-driven approaches. It becomes more challenging to identify and accurately detect these low-frequency attacks, which can potentially evade detection and compromise the security of the network.

The adaptability of IDSs is also a key challenge. The dynamic and flexible nature of networks necessitates regular updates and modifications to IDS models. As the network environment changes, the IDS must adapt to the evolving landscape to maintain its effectiveness. Failure to do so, results in outdated detection models that are ineffective against new and emerging threats. Choosing the appropriate placement strategy for an IDS is another consideration. Organizations must carefully evaluate and select between centralized, distributed, and hybrid deployment strategies based on factors such as financial constraints, computational capabilities, and time costs. Each strategy comes with its advantages and trade-offs, making the decision a critical one. As a final consideration, accuracy poses a significant challenge. Traditional IDS methods often fall short of providing a high degree of precision in detecting intrusions. To address this, a comprehensive and in-depth understanding of intrusion behavior becomes crucial. A complete knowledge of how intrusions manifest and evolve can significantly enhance IDS performance and ensure more accurate threat detection.

In summary, the implementation and improvement of intrusion detection systems face various challenges in today's technological landscape. These challenges include dealing with the diversity of network protocols, addressing low-frequency attacks, ensuring adaptability to changing network environments, making informed placement decisions, and improving accuracy through a deeper understanding of intrusion behavior. Overcoming these challenges requires innovative approaches and continuous research to develop robust and effective IDS solutions.

V. Proposed Method

This section discusses the suggested approach for detecting network intrusions. This section is divided into two subsections, where we explore the requirements and the proposed technique. As depicted in Fig. 2, the suggested NIDS system is divided into three major components.

- The infrastructure layer is composed of two distinct elements: software and hardware. Software elements can communicate with hardware, for example, OpenFlow switches. Hardware elements include switches and routers.
- The control layer regulates activities and data management in the network by creating or refusing each network flow.
- The application layer is responsible for all network management operations. These activities may be accomplished with the aid of a NIDS controller.

As illustrated in Fig. 3, the NIDS generated utilizing machine learning and deep learning algorithms typically entails three key processes, namely data preprocessing, training, and testing. For each of the potential approaches, the dataset is preprocessed and transformed into an algorithm-compatible format. Encoding and normalization are generally included in this step. Frequently, the dataset needs cleaning, which includes eliminating items with duplicate records and incomplete data. The preprocessed data is randomly separated into two parts: the training and testing datasets. Generally, the training dataset accounts for around 80% of the entire dataset, leaving 20% for testing. In the training stage, the deep learning algorithm is trained using the training dataset. The learning time of the method is affected by the complexity of the proposed model and the amount of the dataset. Deep Learning models often need additional training time owing to their deep and complicated underlying structures. After training the model, its performance is evaluated using the testing dataset and its predictions. NIDS models classify network traffic instances as benign (normal) or malicious (attack). The flowchart of the suggested approach is illustrated in Fig. 4. The steps of the proposed technique are described in the following.

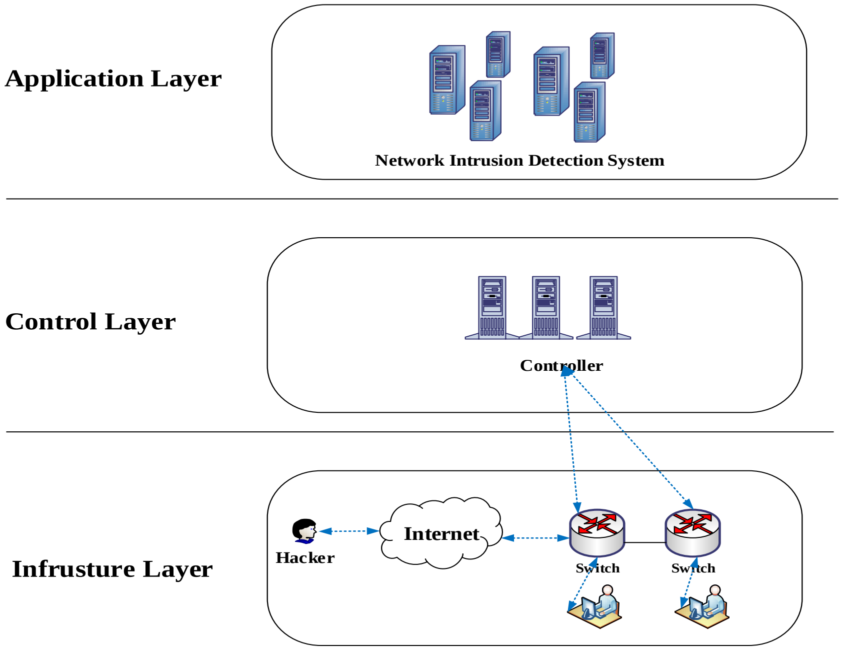

Fig. 2. System model.

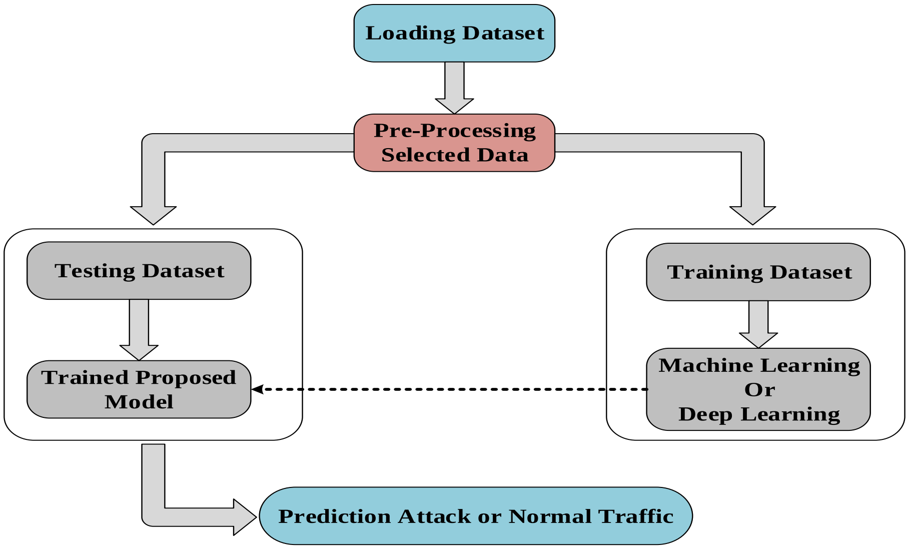
Loading Dataset
Pre-Processing Selected Data
Testing Dataset
Trained Proposed Model
Training Dataset
Machine Learning Or Deep Learning
Prediction Attack or Normal Traffic

Fig. 3. Machine learning and deep learning-based intrusion detection system.

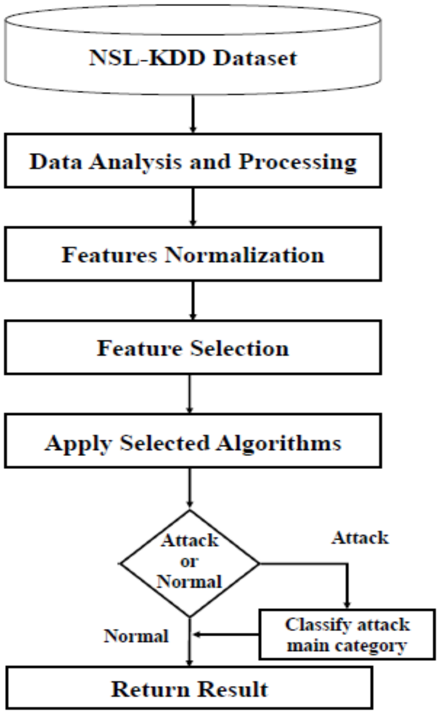

Fig. 4. Flowchart of the proposed NIDS.

A. First Phase: Load Dataset and Export it into Resilient Distributed Datasets (RDDs)

Numerous researchers have developed and evaluated the NIDS issue using the NSL-KDD or other datasets detailed in the assessment part of this proposal. A wide variety of attacks are included in the dataset. It includes 41 features classified into three major categories (traffic-based, content-based, and basic) and distinguished as normal or malicious.

B. Second Phase: Data Preprocessing

The features dataset includes values with varying scale ranges to mitigate the loss function during learning. These scales influence the gradient optimization process, thereby affecting learning rate optimization, as the model should rapidly reach a global or local minimum as a result. Min-Max normalization provides some benefits in comparison to conventional scaling. Min-Max scaling can deal with non-Gaussian feature distributions since anomaly detection applications do not need a certain distribution to follow, in contrast to the signature-based technique in NIDS. The Min-Max normalization strategy is presented to avoid the gradient from the un-smoothing route toward the global minimum, thereby improving the loss function. As illustrated in the following equation, it retrieves the column's lowest and maximum values, with output values ranging from 0 to 1.

$$Normalised\ Parametr\ (X) = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Where X_{min} is the column's lowest value, X_{max} is the column's highest value, and X represents the initial data sample value. Also, we employed the Apache Spark system during this phase. Spark is a high-performance, general-purpose cluster computing system optimized for large-scale in-memory data processing. Spark follows the MapReduce programming paradigm but adds a data-sharing concept called Resilient Distributed Datasets, or RDD. A Spark was developed to be quick for iterative algorithms, enable in-memory storage, and perform well under load.

C. Third phase: Feature Selection

Feature selection forms an integral part of data preprocessing in intrusion detection. A network intrusion detection system is characterized by diverse features and a large amount of data. There are different attribute values for features in different categories, including duplicate features that complicate classification. The proliferation of redundant features reduces the efficiency of detection algorithms and increases the likelihood of false positives in intrusion detection. IDS accuracy and detection speed are increased by an efficient feature selection algorithm, which reduces the dimensionality of network data. This paper uses Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) algorithms for feature selection and training models. CNN is a deep neural network that is composed of three main layers.

- Convolutional layer: In this layer, the input data is processed with a set of filters known as convolutional kernels. A feature map is produced as each filter is applied to the input data. The final output of the convolution layer can be obtained by stacking all the produced feature maps together.
- Pooling layer: It performs subsampling on the feature maps, resulting in reduced dimensionality. The most common methods of pooling are average pooling and maximum pooling.
- Fully connected layer: The output of the previous layers is transformed into a vector that can be used as an input for the next layer.

Recurrent Neural Network (RNN) is a deep learning model driven by supervised learning. Using a traditional RNN, it was possible to predict the temporal training data; however, it encountered difficulties when dealing with gradient explosions. LSTM was proposed as a solution to this problem. An LSTM model replaces the hidden RNN units with a memory function. The LSTM model consisted of three important gates: forget, input, and output gates.

D. Fourth Phase: Train with the Training Dataset

Two causal convolution layers, two dense layers, and a softmax layer are used in the training and optimization phase for the multi-class classification task. In order to avoid overfitting, we employ maximum global pooling, batch normalization, and dropout layers. Adam optimizer is used to update weights and optimize the cross-entropy loss function. It is a combination of two stochastic gradient descent approaches, including Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad). In particular, the training and optimization phase comprises the following layers:

- First causal convolution layer: The input vectors are convolved with 64 filters of three sizes across the input vectors.
- Second 1D causal convolution layer: A total of 128 filters are used, each with a size of 3. Prior to pooling, this layer enables the model to learn more complex features.
- 1D global maximum pooling layer: The maximum value of the filter is replaced with the data covered by the filter. The maximum value prevents the learned features from overfitting.
- Batch normalization layer: The data are normalized before they are sent to the next layer.
- Fully connected dense layer: It utilizes 128 hidden units with a dropout rate of 30%.
- Fully connected dense layer with softmax activation function: multi-class classification is achieved by producing five units for each of the five traffic categories.

VI. Results and Discussion

As choosing the appropriate NIDS data to assess the system is critically important, the data was chosen prior to simulation. While there are a number of publicly available datasets, some of them comprise out-of-date, illogical, inadequately validated, and potentially unrecoverable intrusions. Amazon Web Services (AWS) developed the CSE-CIC-DS2018 [41] dataset to address these limitations and generate modern traffic patterns. It includes a variety of datasets that are suitable for evaluating anomaly-based approaches. CSE-CIC-DS2018 highlights real-time network activities and includes a variety of intrusion detection modes. The data packet payload is calculated by encapsulating the inner network traces as a whole network. Several intrusion profiles are contained in this dataset, which can be applied to a variety of network protocols and topologies. IDS2017 criteria were applied to enhance this dataset. There are currently seven intrusion strategies and two profiles included in IDS2018, a publicly available dataset. IDS2018 contains 80 statistical variables, such as the number of bytes, volume, and packet length. It is accessible via the Internet, which contains approximately 5 million records, and is available in two formats: PCAP and CSV. PCAP is commonly employed to obtain new functions, while the CSV format is typically used in artificial intelligence applications. This dataset represents seven types of attacks: Botnet, Web attacks, Heartbleed, Infiltration, Brute-force SSH, DDOS attacks, and Brute-force DOS attacks.

The creation of the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets has garnered significant interest among researchers, leading to the implementation of various classifiers using these datasets. The datasets' specifications are detailed in Table I. The files within the dataset are utilized for both binary and multi-class classification tasks. An ideal Intrusion Detection System (IDS) is one that can precisely detect each type of attack. To achieve this, building an efficient IDS requires merging the files in the dataset to cover a wide range of attack categories [42]. The CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets exhibit certain limitations related to the data samples and files generated through network flow analysis, which can be listed as follows:

- Tedious data processing: The data samples generated by network flow analysis are stored in files, and processing these files can be a time-consuming and tedious task, especially since each file contains a large number of data instances.
- Dataset size and computing time: Merging the files in the dataset to include all attack labels can lead to an increase in the dataset's size. This, in turn, results in more computing and processing time, making it challenging to handle large datasets efficiently.
- Missing and redundant data: The dataset contains some missing and redundant data records, which can affect the quality and accuracy of the analysis performed on the data.
- High-class imbalance: Both CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets suffer from the issue of high-class imbalance. This means that some attack types may

have significantly fewer instances compared to others, leading to lower accuracy and higher False Positive Rate (FPR) for the system.

There are 50 computers involved in the dataset-attacking infrastructure, while 30 servers and 420 terminals are utilized by the attackers. CSE-CIC-DS2018 data represent a system log with 80 attributes extracted from CICFlowMeter-V3 and network traffic captured from AWS. There is approximately 400 GB of data in CSE-CIC-DS2018, which is larger than CIC-DS2017 in terms of size. Table II compares the CSE-CIC-DS2018 and CIC-DS2017 datasets with respect to sample size. The number of CSE-CIC-DS2018 samples has increased significantly compared to CIC-DS2017, especially in the Infiltration and Botnet attacks, where the number of samples increased respectively by 4497 and 143.

Using the CSE-CIC-DS2018 ID dataset, the effectiveness of our mechanism was evaluated by examining error rate, accuracy, true negative, false negative, true positive, and false positive. The confusion matrix is used to determine the difference between the actual and predicted classifications. Categorization results can be divided into two groups: normal and abnormal. Table III provides an overview of the confusion matrix. It is necessary to measure four levels of criticality in the confusion matrix.

- True negative: In this case, the model correctly predicts the negative outcome.
- False positive: In this case, the classifier considers normal traffic as abnormal.
- False negative: When an IDS fails to detect an actual attack.
- True positive: It is an actual intrusion successfully discovered by the IDS.

Based on the specifications given above for the confusion matrix, we can calculate the output of the system. An IDS is analyzed based on FAR and DR as key and common metrics. FAR represents the sum of misclassified regular incidents, whereas DR signifies the number of intrusions identified by the model. As DR rises and FAR decreases, we claim our approach is superior to traditional approaches.

$$FAR = {}^{FP}/_{(TN + FP)} \quad (2)$$

$$DR = {}^{TP}/_{(TP + FN)} \quad (3)$$

The performance of the classifier on CSE-CIC-DS2018 is summarized in Table IV. A random search hyperparameter optimization technique was used to generate the results. The ensemble classifier XGB significantly enhances classification effectiveness, achieving an accuracy rate of 85%. The tree-based classifier provides a higher level of accuracy than ensemble-based classifiers.

TABLE I. SPECIFICATIONS OF CIC-IDS-2017 AND CSE-CIC-IDS-2018 DATASETS

Dataset	Type	Number of classes	Features	Victim Infrastructure	Attack Infrastructure	Duration of Capture
CSE-CIC-DS2018	Multi-class	18	80	420 PCs, 30 servers	50 PCs	Ten days
CIC-DS2017	Multi-class	15	80	Three server, one firewall, two switches, 10 PCs	Four PCs, one router, one switch	Five days

TABLE II. COMPARISON OF THE CSE-CIC-DS2018 ID DATASET WITH CIC-DS2017

Dataset	Web attacks	Infiltration	Brute force	Botnet	DoS	DDoS	Normal
CSE-CIC-DS2018	929	161,936	380,950	286,195	954,311	687,740	6,112,149
CIC-DS2017	2182	37	13,840	1968	252,665	128,024	1,743,181

TABLE III. OVERVIEW OF THE CONFUSION MATRIX

	Predicted outcome		
Actual value	Abnormal	True negative	False positive
	Normal	False negative	True positive

TABLE IV. CLASSIFIER RESULTS WITH CSE-CIC-DS2018

Classifier	FAR	DR	F-score	Recall	Precision
DT	7.81	0.89	0.87	0.88	0.87
XGB	9.1	0.84	0.83	0.83	0.84
LR	11.5	0.80	0.79	0.80	0.78
Proposed classifier	2.6	0.96	0.981	0.976	0.968

TABLE V. COMPARISON OF IDS METHODS

References	False alarm rate	Accuracy
[43]	1.1	96%
[44]	1.3	96%
[45]	8.5	96%
[46]	5	95%
[41]	0.93	90.2%
[47]	0.97	94%
Our method	1.5	98.2%

In order to use the proposed mechanism to capture both temporal and spatial features efficiently, recurrent layers were introduced following the CNN layers in order to capture both features. In this manner, we attempted to avoid the vanishing gradient and explosion problem, resulting in an improved ability to capture temporal and spatial dependencies and learn efficiently from sequences of variable extent. The number of variables in large-scale data (imbalances), while exceeding the sample size, is not well suited to traditional machine-learning classifiers. This model is suitable for high-dimensional datasets due to its scale invariance. Nevertheless, the most significant improvement was achieved with advanced deep learning approaches such as CNNRNN, which detected misuse with 97% accuracy. This performance improvement can be attributed to the long-term dependencies between the nonlinear features, and details on their implementation can be found in supplementary materials.

A summary of the results obtained using existing methods for the CSE-CIC-DS2018 dataset is provided in Table V. Several preliminary results are available since these datasets are generated following the KDD and DARPA datasets. Considering current simulation results, optimal values for accuracy and FAR was calculated for each phase. The accuracy and FAR of our method are superior to those of conventional methods. The reason for this is the execution of the deep learning algorithm. Due to the differences in the quantity of data distributions, preprocessing procedures, and sampling methods, the similarities should only be used as a source of reference. Therefore, measuring a simple metric, such as the amount of time spent on testing or training is rarely appropriate. Although the suggested method demonstrated superior performance in some respects, it remains questionable whether it can perform better in all respects than other approaches. The proposed solution enables exceptional network protection as well as easy identification of malicious threats.

In comparison to previous approaches, the proposed method demonstrates superiority based on simulation results. It achieves an exceptional level of accuracy in detecting malicious attacks, as evidenced by its outstanding performance on the widely recognized CSE-CIC-DS2018 dataset. This indicates that the proposed NIDS has surpassed the capabilities of existing methods in terms of accuracy and effectiveness. By capturing packets from the network and performing careful preprocessing to reduce their size while retaining crucial information, the proposed method optimizes the input data for the deep learning algorithm. The utilization of machine learning characteristics enhances the NIDS's ability to learn and test potential intrusion patterns, further contributing to its superior performance.

The hybrid feature selection approach introduced in the proposed method addresses the challenge of identifying critical attributes necessary for constructing robust classifiers. This feature selection process, combined with rule assessment and detection steps, provides a comprehensive and effective framework for intrusion detection. In summary, the proposed method stands out among previous approaches by leveraging deep learning, hybrid feature selection, and a carefully designed process flow. Its exceptional accuracy in detecting malicious attacks on the CSE-CIC-DS2018 dataset indicates its superiority over existing methods. The proposed method offers significant advancements in network intrusion detection and presents a promising solution to bolster the security of networks and systems in today's digital landscape.

VII. Conclusion

Cyber security has become a paramount area of research in modern society, given the indispensable role of networks. Within this domain, Intrusion Detection Systems (IDSs) play a pivotal role in monitoring the status of software and hardware on a network. However, IDSs continue to face challenges in accurately identifying potential threats, minimizing false alarms, and enhancing detection accuracy. To address these challenges, extensive research has been dedicated to developing IDSs that harness the power of machine learning. In our study, we have devised a cutting-edge IDS methodology based on a layered Recurrent Neural Network (RNN). This approach leverages the strengths of deep learning to adeptly predict and classify unauthorized intrusions. The layered RNN effectively captures local features, while the recurrent RNN seamlessly incorporates temporal characteristics, substantially elevating the performance of our IDS system. Through comprehensive evaluations on the esteemed CSE-CIC-DS2018 dataset, our proposed method has demonstrated superior performance over previous approaches. The simulation results unequivocally establish the exceptional accuracy of our IDS in detecting malicious attacks. The integration of deep learning techniques and hybrid feature selection enables our IDS, named DeepShield, to outperform traditional machine learning-based methods, providing a reliable defense against network intrusions. The significance of DeepShield extends beyond the system itself. The higher accuracy achieved by our NIDS translates to more robust network security, helping organizations proactively safeguard their critical data and online services. As cyber threats continue to evolve, the effectiveness of intrusion detection becomes increasingly crucial, and our approach contributes to a safer digital environment. Moreover, the versatility of our methodology allows for scalability and adaptability to various network infrastructures and environments. This adaptability ensures that DeepShield can be applied in diverse security scenarios, making it a valuable tool for network administrators and cyber security professionals.

References

[1] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[2] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9326-9337, 2019.

[3] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic," Peer-to-Peer Networking and Applications, pp. 1-21, 2022.

[4] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023.

[5] H. Tao et al., "Economic perspective analysis of protecting big data security and privacy," Future Generation Computer Systems, vol. 98, pp. 660-671, 2019.

[6] S. Quintero-Bonilla and A. Martín del Rey, "A new proposal on the advanced persistent threat: A survey," Applied Sciences, vol. 10, no. 11, p. 3874, 2020.

[7] W. Chen et al., "Advanced persistent threat organization identification based on software gene of malware," Transactions on Emerging Telecommunications Technologies, vol. 31, no. 12, p. e3884, 2020.

[8] R. P. Baksi and S. J. Upadhyaya, "Decepticon: a Theoretical Framework to Counter Advanced Persistent Threats," Information Systems Frontiers, vol. 23, no. 4, pp. 897-913, 2021.

[9] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

[10] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," Information, vol. 7, no. 2, p. 25, 2016.

[11] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," Knowledge-Based Systems, vol. 189, p. 105124, 2020.

[12] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," Journal of Big Data, vol. 7, no. 1, pp. 1-20, 2020.

[13] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," Computers & Security, vol. 92, p. 101752, 2020.

[14] R. Soleimani and E. Lobaton, "Enhancing Inference on Physiological and Kinematic Periodic Signals via Phase-Based Interpretability and Multi-Task Learning," Information, vol. 13, no. 7, p. 326, 2022.

[15] B. M. Jafari, M. Zhao, and A. Jafari, "Rumi: An Intelligent Agent Enhancing Learning Management Systems Using Machine Learning Techniques," Journal of Software Engineering and Applications, vol. 15, no. 9, pp. 325-343, 2022.

[16] M. Bagheri et al., "Data conditioning and forecasting methodology using machine learning on production data for a well pad," in Offshore Technology Conference, 2020: OTC, p. D031S037R002.

[17] S. R. Abdul Samad et al., "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," Electronics, vol. 12, no. 7, p. 1642, 2023.

[18] S. Aghakhani, A. Larijani, F. Sadeghi, D. Martín, and A. A. Shahrakht, "A Novel Hybrid Artificial Bee Colony-Based Deep Convolutional Neural Network to Improve the Detection Performance of Backscatter Communication Systems," Electronics, vol. 12, no. 10, p. 2263, 2023.

[19] B. M. Jafari, X. Luo, and A. Jafari, "Unsupervised Keyword Extraction for Hashtag Recommendation in Social Media," in The International FLAIRS Conference Proceedings, 2023, vol. 36.

[20] C. Han and X. Fu, "Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model," Frontiers in Business, Economics and Management, vol. 8, no. 2, pp. 51-54, 2023.

[21] M. Javidan, M. Yazdchi, Z. Baharlouei, and A. Mahnam, "Feature and channel selection for designing a regression-based continuous-variable emotion recognition system with two EEG channels," Biomedical Signal Processing and Control, vol. 70, p. 102979, 2021.

[22] G. Shen, W. Zeng, C. Han, P. Liu, and Y. Zhang, "Determination of the average maintenance time of CNC machine tools based on type II failure correlation," Eksploatacja i Niezawodność, vol. 19, no. 4, 2017.

[23] M. Shahin et al., "Cluster-based association rule mining for an intersection accident dataset," in 2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2021: IEEE, pp. 1-6.

[24] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," Simulation Modelling Practice and Theory, vol. 101, p. 102031, 2020.

[25] M. Farooq, "Supervised Learning Techniques for Intrusion Detection System based on Multi-layer Classification Approach," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, 2022.

[26] A. Iqbal and S. Aftab, "A Feed-Forward and Pattern Recognition ANN Model for Network Intrusion Detection," International Journal of Computer Network & Information Security, vol. 11, no. 4, 2019.

[27] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, p. e4150, 2021.

[28] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," Expert Systems with Applications, vol. 186, p. 115782, 2021.

[29] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems," in International Conference on Big Data Technologies and Applications, International Wireless Internet Conference, 2021: Springer, pp. 117-135.

[30] S. Vairachilai, A. Bostani, A. Mehbodniya, J. L. Webber, O. Hemakesavulu, and P. Vijayakumar, "Body Sensor 5 G Networks Utilising Deep Learning Architectures for Emotion Detection Based On EEG Signal Processing," Optik, p. 170469, 2022.

[31] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," Cluster Computing, vol. 23, no. 2, pp. 1397-1418, 2020.

[32] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," Ieee Access, vol. 5, pp. 21954-21961, 2017.

[33] H. M. Anwer, M. Farouk, and A. Abdel-Hamid, "A framework for efficient network anomaly intrusion detection with features selection," in 2018 9th International Conference on Information and Communication Systems (ICICS), 2018: IEEE, pp. 157-162.

[34] Y. Tian, M. Mirzabagheri, S. M. H. Bamakan, H. Wang, and Q. Qu, "Ramp loss one-class support vector machine; a robust and effective approach to anomaly detection problems," Neurocomputing, vol. 310, pp. 223-235, 2018.

[35] D. A. Effendy, K. Kusrini, and S. Sudarmawan, "Classification of intrusion detection system (IDS) based on computer network," in 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017: IEEE, pp. 90-94.

[36] X. Kan et al., "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," Information Sciences, vol. 568, pp. 147-162, 2021.

[37] G. Andresini, A. Appice, and D. Malerba, "Autoencoder-based deep metric learning for network intrusion detection," Information Sciences, vol. 569, pp. 706-727, 2021.

[38] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," Computers and Electrical Engineering, vol. 102, p. 108156, 2022.

[39] M. A. Talukder et al., "A dependable hybrid machine learning model for network intrusion detection," Journal of Information Security and Applications, vol. 72, p. 103405, 2023.

[40] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," International Journal of Information Security, vol. 22, no. 1, pp. 235-247, 2023.

[41] R. I. Farhan, A. T. Maolood, and N. F. Hassan, "Optimized Deep Learning with Binary PSO for Intrusion Detection on CSE-CIC-IDS2018 Dataset," Journal of Al-Qadisiyah for computer science and mathematics, vol. 12, no. 3, pp. Page 16-27, 2020.

[42] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," International Journal of Engineering & Technology, vol. 7, no. 3.24, pp. 479-482, 2018.

[43] J. Kim, Y. Shin, and E. Choi, "An intrusion detection model based on a convolutional neural network," Journal of Multimedia Information System, vol. 6, no. 4, pp. 165-172, 2019.

[44] Q. Zhou and D. Pezaros, "Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection--An Analysis on CIC-AWS-2018 dataset," arXiv preprint arXiv:1905.03685, 2019.

[45] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," in International conference on cloud computing, 2019: Springer, pp. 161-176.

[46] R. I. Farhan, A. T. Maolood, and N. Hassan, "Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning," Indones. J. Electr. Eng. Comput. Sci, vol. 20, no. 3, pp. 1413-1418, 2020.

[47] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, "An optimization method for intrusion detection classification model based on deep belief network," Ieee Access, vol. 7, pp. 87593-87605, 2019.