**Hybrid Approach to Packet Sniffing and Anomaly Detection Using Deep Learning**

Ammar Mohammed\**a,b*, Mohamed Amin *a*, Hesham Hefny *a*

*aDepartment of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt*

*bDepartment of Computer Science, Misr International University, Cairo, Egypt*

\*Corresponding Author: Ammar Mohammed [***ammar@cu.edu.eg***]

ARTICLE DATA ABSTRACT

|  |  |  |
| --- | --- | --- |
| ***Article history:***  ***Received 27 April 2022***  ***Revised 08 August 2022***  ***Accepted 09 August 2022***  ***Available online*** |  | In the recent past, cyber threats have increased significantly due to the number of vulnerabilities in many systems. This has posed more challenges to Network Intrusion Detection Systems (NIDS). Some of them include an overwhelming amount of network traffic that IoT devices face, most of those systems find a problem achieving high accuracy in detecting those vulnerabilities and identifying low impact but infrequently occurring attacks. This paper presents a hybrid technique where the proposed deep learning model is capable of reinforcement learning with Feedback to solve the above-mentioned problems. Our system is designed to combine an in-depth analysis of packet payload with an adaptive anomalous detection mechanism. Results demonstrate the effectiveness of technology in solving important critical challenges leading to cybersecurity solutions. |
| *Keywords: NIDS, RL, FSM, CNN, 1D-CNN, LSTM, KNN.* |  |
|  |  |

# 1. Introduction

The internet has many systems and services, providing a lot of convenience and advancement in securing those systems. However, as it grows, vulnerability complexity increasing, introduces a need to face those threats against the security systems, transforming itself from massive attacks to high complex ones, low frequency breaches. Such threats exploit vulnerabilities within modern networks to rise against organizations as well as individuals. Hence, Network Intrusion Detection Systems (NIDS) have been designed to form the core of their defense against these attacks. They oversee an important function-the monitoring and analysis of network traffic with ulterior motives attached.

Our goal was only to test it against one of the most commonly known attacks, SSH (Secure Shell) brute frorce attack which the attacker is repeatedly trying username and password combinantions unti the attacker gains access to the remote server. SSH brute force attack is effective and very fast way to compromise a server.

Most of the challenges faced by conventional NIDS, particularly in an IoT environment where human life relies on the security of these systems. Conditions of highly sensitive exhibit less capability to achieve detection rates that are at par, for many systems seem incapable of reaching the accuracy level necessary for identifying minor scale yet critical attacks. It happens because conventional systems could not develop a methodology for adaptation to new shapes of attack and how meaningful features would be extracted from the complicated high dimensional data. Thus, research and development in the above area will need a comprehensive approach in using the latest technologies to improve the accuracy and robustness of detection.

For this reason, we present a model somehow deep hybrid learning-based to solve those challenges above.

Using the effectiveness of deep learning and to increase accuracy rapidatly.Along side to a framework for Reinforcement Learning (RL), next to feedback mechanisms, is being incorporated to adapt the system dynamically to changes in network behavior and attack strategies.

# 2. Background

In this section, we delve into the challenges faced by modern NIDS and how deep learning can help overcome these challenges, finally the approaches combining NIDS and deep learning all together.

### Challenges of NIDS

Network Intrusion Detection Systems (NIDS) encounter various challenges that are often overlooked yet essential for their success. One significant issue is the data imbalance frequently found in training datasets, where attacks occur much less often than normal traffic. This imbalance can result in high false positive rates or inadequate detection of rare but critical attacks. Another concern is the susceptibility of NIDS to adversarial attacks, where slight alterations in network traffic can deceive the system into misclassifying harmful traffic as harmless. This problem is becoming increasingly relevant as the field of cybersecurity advances. Additionally, the high dimensionality of network traffic data complicates the task of preserving vital information while reducing features. This challenge intensifies when employing deep learning methods, as it is crucial to select the right features without losing important attack signatures. Furthermore, latency in real-time detection can pose a problem, particularly in hardware-accelerated systems like FPGA, where there is often a trade-off between model complexity and performance. Lastly, NIDS must also address the detection of zero-day and novel attacks, a challenge that is not often discussed but is essential for providing strong protection against emerging threats.

**Accuracy**- Based on some of these levels of accuracy, it does not mean that one can rely on existing techniques. Therefore, more granularity, deeper and contextual understanding are needed in order to give a more holistic picture that is, however, costly, in terms of money, computational power, and time. [5]

**Diversity**- More than before there has been a rise in the number of new or customized protocols in modern networks, due in part to the number of devices that either connect to a network or go online to the Internet. Consequently, the very definition of normal and abnormal traffic and/or behaviors becomes increasingly complicated [6].

**Dynamics**- Because of the different installations and flexibility of the current day networks, the behavior is dynamic and hard to predict, which in turn leads to a big challenge in defining a reliable behavioral norm.

**Adaptability**- Most modern networks have adopted several new technologies to break from previous static technologies and modern management styles. Thus, there is more proliferation of dynamic technologies like containerization, virtualization and Software Defined Networks. This, in turn, requires NIDSs to adapt to such technologies and the side effects they produce.

### *Role of Deep Learning*

# Deep learning has been at the core of the advancements in machine learning in the recent past, bringing AI closer to being. This domain is excellent in modeling relations of complexity and finding meaningful abstractions through hierarchical levels of representation. Using supervised and unsupervised algorithms, it makes high-level constructs from built-in lower-level constructs stratifying great models for intricate data distributions.

Traditional intrusion detection techniques using signatures and rules often suffer with regards to their effectiveness in discovering changing attack techniques, especially when adversaries use evasion techniques through obfuscation to evade detection. Deep learning technology, employing architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers, have the capability to inspect complex network traffic, extract meaningful features, and detect both traditional and zero-day attacks with less reliance on human intervention.

A key use case for deep learning in Network Intrusion Detection Systems (NIDS) is in identifying SSH brute-force attacks, a common attack in which attackers try to gain unauthorized access through a systematic testing of username and password combinations [1]. Traditional techniques rely on threshold approaches, such as tracking failed logins in a specific period of time. However, such approaches can cause high false positive rates or miss slow-rate brute-force attacks that make login attempts over a long duration in an attempt to evade detection. Deep learning algorithms overcome such weaknesses through distinguishing complex spatio-temporal and temporal relationships in network traffic information, allowing for distinguishing between normal logins and suspicious brute-force activity.

For discovering SSH brute-force attacks, deep learning algorithms inspect a range of network flow features, such as request attempt frequencies, session duration, packet sizes, and request interval times. Sequence-based algorithms, such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs), work best in discovering spatio-temporal dependencies, discovering anomalous behavior during the authentication stage, and discovering attack signatures even when attackers use distributed and covert techniques. Hybrid deep architectures combining CNNs with RL make feature extraction easier and allow for spatio-temporal feature extraction, enhancing accuracy in detection.

# 3. Related Work

Network Intrusion Detection Systems (NIDS) are crucial for securing networked environments, and their study has gained significant attention. As cyberattacks become more sophisticated, the demand for effective and adaptable NIDS has intensified. A considerable amount of research has been dedicated to leveraging machine learning (ML) and deep learning (DL) techniques to enhance NIDS performance, particularly for real-time intrusion detection. Traditional methods, such as signature-based and anomaly-based detection, face challenges in addressing the ever-evolving landscape of cyber threats. Signature-based approaches can only recognize known attack patterns, while anomaly-based methods often experience high false positive rates due to the fluctuating nature of network traffic. In recent years, numerous studies have investigated the use of deep learning models for NIDS, as these models can automatically extract features from raw network traffic, leading to improved accuracy in identifying complex attacks. For example, Shone et al. (2018) [1]introduced a deep learning method utilizing a deep neural network (DNN) for NIDS, showcasing better performance compared to traditional machine learning techniques (Shone et al., 2018). Their research indicated that deep learning models, especially convolutional neural networks (CNNs), can automatically learn spatial hierarchies of network features, thereby enhancing classification accuracy. Likewise, Zhang et al. (2020) [2]presented IDSNet, a deep learning framework that combines multiple layers of learning to boost detection accuracy and resilience against adversarial attacks (Zhang et al., 2020). effectiveness of deep learning models in identifying sophisticated threats, particularly in situations where conventional models fall short.

A significant area of research focuses on the development of lightweight deep learning models for network intrusion detection systems (NIDS). This effort aims to lower computational demands and facilitate real-time intrusion detection, particularly in environments with limited resources, such as embedded systems and Internet of Things (IoT) networks. For instance, [1] introduced a lightweight framework for intrusion detection in wireless sensor networks, highlighting the necessity of model efficiency for real-time applications. Their approach effectively reduces computational load while ensuring a high detection rate. Similarly, Niyaz et al. (2016) [2] created a deep learning-based system for detecting DDoS attacks that utilized software-defined networking (SDN) to enhance resource allocation and minimize latency. Their findings indicated that the combination of deep learning and SDN can significantly improve real-time threat detection in high-traffic environments. Beyond deep learning, another vital research area is the detection of zero-day attacks. The difficulty of identifying previously unknown threats has sparked interest in unsupervised learning methods and hybrid models that merge signature-based and anomaly-based techniques. Sharafaldin et al. (2018) [3] focused on generating new datasets for training intrusion detection systems, which included attacks not represented in established datasets like KDD99. This advancement allowed for more precise detection of novel and zero-day attacks. Likewise, McHugh (2000) [4]emphasized the necessity of testing intrusion detection systems under diverse conditions to ensure their capability to identify unknown threats and adapt to emerging attack strategies.

Adversarial attacks on Network Intrusion Detection Systems (NIDS) are becoming an increasing concern. With the rise of deep learning models in network security, their vulnerability to adversarial manipulation is a significant issue. LeCun et al. (2015) [5] highlighted how deep learning models can be misled by small, carefully designed inputs that lead to misclassification, often without detection by traditional security measures. This challenge has spurred recent research aimed at enhancing the robustness of NIDS against such attacks. One promising strategy is adversarial training, where the model learns from adversarial examples during its training phase, thereby boosting its resilience. Nonetheless, this field is still developing, as effective defense methods against adversarial attacks continue to evolve. In summary, the research landscape for NIDS has made notable progress, especially with the adoption of deep learning techniques that enhance detection accuracy, minimize false positives, and improve the identification of new attack vectors. However, challenges persist in real-time deployment, particularly in high-latency settings and under adversarial conditions. Ongoing research aims to tackle these challenges while enhancing the efficiency and accuracy of NIDS to meet the demands of modern network security.

# 4. Proposed System Architechture

The proposed system architecture is designed to enable real-time, high-performance network intrusion detection by using a layered approach that ensures modularity, scalability, and efficient data flow. By using deep learning-based classification, the system can swiftly detect SSH brute force attacks, with high accuracy. Each layer is structured to handle specific tasks, from data acquisition to alert reporting, creating a seamless and optimized intrusion detection pipeline.

### Layered Architecture Approach

The proposed system architecture adopts a layered approach to achieve real-time, high-performance network intrusion detection through deep learning. This design promotes modularity, scalability, and efficient data flow while optimizing computational resources. The architecture is composed of six essential layers: Data Acquisition Layer, Data Processing Layer, Intrusion Detection Layer, Alert & Reporting Layer, Management & Control Layer, and Database Layer. Each layer has specific responsibilities, contributing to a comprehensive intrusion detection pipeline that effectively identifies cyber threats, particularly SSH brute-force attacks, in real-time.

The Data Acquisition Layer acts as the system's foundation, capturing live network traffic with a packet sniffer and forwarding it to traffic aggregators. This aggregation process ensures that related packets are grouped together, creating meaningful sequences that enhance analysis in subsequent stages. By organizing the raw network data in this way, the system preserves vital information while minimizing redundant data transmission.

The Data Processing Layer transforms raw network traffic into structured features suitable for deep learning analysis. This layer includes a feature extraction module that identifies relevant attributes from network packets, such as connection attempts, protocol types, session durations, and timing intervals. The extracted features are normalized to ensure consistency in input representation, preventing bias from varying data scales. The processed data is temporarily stored in a buffer, maintaining a steady data flow to the detection engine and avoiding bottlenecks in real-time analysis.

The Intrusion Detection Layer serves as the system's foundation, integrating FPGA-based acceleration with deep learning models to achieve fast and accurate intrusion detection. The FPGA inference engine handles the pre-extracted features and enhances the deep learning model's performance, enabling quick classification of network traffic. This model, trained on a labeled dataset that includes both normal and malicious network behaviors, identifies SSH brute-force attacks and other cyber threats by recognizing learned patterns. The classification module sorts incoming network traffic into either normal or suspicious categories, forwarding the findings to the next layer for reporting and response.

The Alert & Reporting Layer plays a crucial role in alerting system administrators about detected threats and creating comprehensive incident reports. The alert system promptly issues warnings when suspicious activity is detected, facilitating immediate threat response. At the same time, the report generator assembles structured reports that detail the detected attack, including timestamps, source and destination IP addresses, attack severity, and behavioral patterns. The user interface features an intuitive dashboard that allows administrators to review alerts, analyze trends, and adjust detection settings as necessary.

The Management & Control Layer grants system administrators full oversight of the NIDS. This layer encompasses configuration settings that enable the customization of detection thresholds, model parameters, and alert preferences. The system monitor (dashboard) offers real-time visualizations of system performance, detected threats, and network traffic analytics. Authorized personnel can access admin features to supervise system operations, update detection models, and manage security policies.

The Database Layer plays a crucial role in storing and managing important system data. It features an anomaly log database that records detected security incidents for later analysis. The system metrics database keeps track of performance logs, such as processing times, detection accuracy, and resource usage, which helps in ongoing optimization. Additionally, the training data repository holds labeled datasets that are used to train and refine the deep learning model, facilitating continuous enhancement of attack detection capabilities.

A diagram of a system

Description automatically generated

Figure 1 Proposed system Architecture

## 4.2. 1D-CNN Architechture

Figure 2 shows how 1D-CNN architecture is advantaged to other commonly used deep learning and machine learning models when are used in systems along side to RL.

### Pattern Recognition Beyond Traditional Methods

1D CNNs stand out from traditional rule-based or statistical anomaly detection methods by automatically identifying attack signatures without needing prior domain knowledge [8]. This feature is especially beneficial for spotting zero-day attacks and adversarial traffic, where attack behaviors are constantly changing.

### Resilience to Feature Engineering Bias

In conventional NIDS methods, feature selection can introduce bias since it depends on manually selected features based on past attack patterns. In contrast, 1D CNNs dynamically learn the best feature representations, making them more flexible in responding to new and complex cyber threats. [9]

### Effective SSH Brute-Force Attack Detection

Temporal Pattern Learning: 1D CNNs identify SSH brute-force attacks by detecting irregularities in authentication attempts over time. Unlike signature-based IDS that depend on predefined attack fingerprints, CNNs can recognize new variations of brute-force attacks by learning patterns from raw traffic.

High Precision and Low False Positives: By training on both attack and normal traffic datasets (e.g., KDD 99, CIC-IDS 2017), 1D CNNs achieve high detection accuracy while keeping false alarms to a minimum.

A diagram of a agent and environment

Description automatically generated

# 5. Proposed Algorithm

To achieve real-time network intrusion detection with high accuracy and efficiency, we propose a deep learning-based approach leveraging a one-dimensional convolutional neural network (1D-CNN). This section details the core components of our proposed method.

## 5.1. 1D-CNN

A 1D Convolutional Neural Networks (1D CNN) is a specialized deep learning architecture designed to extract meaningful spatial patterns from sequential data. Unlike 2D CNNs, which are commonly used for image processing, 1D CNNs operate on one-dimensional sequential data, making them particularly effective for analyzing network traffic, time-sesiteve data, and log sequences into our intrusion detection system (IDS). The architecture consists of main 6 key layers, each playing an important role in feature extraction, transformation, and classification. We will break those layers to help explain the process and role of each layer.

#### Input Layer

The input to a 1D CNN in a Network Intrusion Detection System (NIDS) is typically a sequence of numerical features extracted from network traffic, such as packet sizes, inter-arrival times, protocol types, and byte distributions. These features are fed as a 1D vector, where each dimension corresponds to a particular network characteristic over time.

#### Convolutional Layer

The convolutional layer applies 1D filters (kernels) that slide over the input sequence, capturing local dependencies and hierarchical relationships between features. Unlike traditional feature selection methods that rely on domain expertise, CNNs learn feature representations automatically, identifying subtle attack patterns that might be overlooked by conventional approaches.

Receptive Field Adaptability: The receptive field of 1D CNNs is flexible and depends on kernel size and stride. Larger kernels can capture long-term dependencies in network traffic, which is critical for detecting slow and distributed attacks, while smaller kernels focus on fine-grained anomalies.

Weight Sharing and Computational Efficiency: Since the same filter is applied across the input sequence, 1D CNNs reduce computational overhead compared to fully connected architectures, making them highly efficient for real-time applications on FPGA hardware.

#### Pooling Layer

Pooling operations, such as max pooling or average pooling, reduce dimensionality while retaining the most important features. This step prevents overfitting, enhances generalization to unseen attacks, and ensures computational efficiency by reducing the number of trainable parameters.

Temporal Reduction: By summarizing segments of the network traffic data, pooling layers help capture attack patterns spread across time (e.g., slow brute-force attacks).

Noise Reduction: Pooling layers eliminate minor variations in traffic data, focusing on the most distinctive attack features.

#### Flatten Layer

This layer reshapes the multi-dimensional feature maps from the convolutional and pooling layers into a 1D vector. This transformation enables the transition from feature extraction to classification.

#### Fully Connected Layer

Fully connected layers learn complex decision boundaries by mapping extracted features to output classes. Each neuron in this layer represents a learned combination of network traffic features that correlate with normal or anomalous behaviors.

Activation Functions (e.g., ReLU, Leaky ReLU): These introduce non-linearity, allowing the model to capture intricate patterns beyond simple linear relationships.

Dropout for Regularization: To mitigate overfitting, dropout techniques can be applied, randomly deactivating neurons during training to enhance model robustness.

#### Output Layer

The final layer produces a probability distribution over attack classes using activation functions such as softmax (for multi-class classification) or sigmoid (for binary intrusion detection). In the case of SSH brute-force detection, the model would classify input sequences as either normal traffic or an attack.

Figure 2: 1D-CNN Layers

## 5.2. Dataset

The CICIDS 2018 dataset is the most comprehensive and realistic benchmark for evaluating IDS, especially for machine learning and deep learning models. Developed by the Canadian Institute for Cybersecurity (CIC), this dataset was designed to overcome the limitations of older datasets like KDD99 and NSL-KDD which have outdated attack patterns, redundant records and lack of real-world traffic diversity. CICIDS 2018 was collected in a controlled yet highly realistic environment, simulating modern network traffic and attacks while maintaining traffic flow characteristics. This dataset is a goldmine for cybersecurity research, with labeled data for both normal and malicious traffic, covering a wide range of contemporary attack types including SSH brute-force, botnets, DoS, web attacks

### Dataset Background

The dataset was collected by simulating an enterprise network over a period of 10 days, capturing real-world attack scenarios in a structured way. This network had multiple victim machines running different operating systems (Windows, Linux, macOS), legitimate users performing routine activities (browsing, emailing, file transfers) and an adversarial component executing manual and automated cyber-attacks. One of the unique features of this dataset is that it includes traffic from Tor network based attacks, encrypted malicious activity and anonymized traffic, making it more challenging and realistic than older datasets.

Each network flow in CICIDS 2018 was collected using CICFlowMeter, a tool that extracts 80 network traffic features, including packet size distributions, inter-arrival times, byte counts, flow durations and protocol based behaviors. This level of granularity is very useful for deep learning based anomaly detection, as it allows models to learn complex temporal patterns in network traffic.

### CICIDS 2018 Valuable for SSH Brute-Force Detection

Among the various attack types recorded in this dataset, SSH brute-force attacks stand out as particularly critical. These attacks involve repeated login attempts to a secure shell (SSH) server, utilizing automated scripts that try different combinations of usernames and passwords. Unlike other brute-force datasets that merely log failed login attempts, CICIDS 2018 offers a complete packet capture (PCAP) of the attack sessions, enabling a thorough examination of attack behaviors, such as:

Irregular authentication attempts: A pattern of repeated failed logins followed by an unexpected success.

High-frequency connection attempts: Attackers often rotate through various IP addresses or employ botnets to spread out their login attempts [11].

Protocol-specific anomalies: While legitimate SSH traffic tends to follow predictable patterns, brute-force attacks create statistically significant deviations in packet inter-arrival times and payload structures.

One of the most unique and informative features of CICIDS 2018 is its capacity to reveal malicious port usage that strays from typical network behaviors. For instance:

Destination port 22 is the standard SSH port, and brute-force attacks usually involve quick authentication attempts on this port. However, CICIDS 2018 shows that attackers frequently target non-standard ports to avoid detection.

Destination port 53, generally used for DNS queries, has been noted in some attack scenarios, suggesting possible DNS tunneling techniques where SSH brute-force attempts are hidden within what appears to be normal DNS traffic [11].

Unexpected use of ephemeral ports: SSH brute-force tools may sometimes connect through high-numbered ephemeral ports (e.g., 49152–65535) to circumvent static firewall rules.

Traditional intrusion detection systems face challenges with stealthy brute-force attacks, where login attempts are spread out over long periods to evade standard rate-limiting measures. The CICIDS 2018 dataset offers time-series attack data [11], allowing deep learning models (like 1D CNNs and LSTMs) to examine:

* The gradual increase in login attempts over time.
* The shift from failed logins to successful breaches, which could suggest credential stuffing or compromised credentials.
* Changes in response packet sizes, which may indicate whether a system is sending different failure messages based on the attacker's progress.

Additionally, CICIDS 2018 features a variety of attack vectors, including:

* Dictionary-based SSH brute-force (utilizing a predefined list of credentials).
* Intelligent brute-force (where attack tools modify their tactics based on server feedback).
* Distributed brute-force (employing multiple IP addresses to distribute login attempts, complicating detection).

This dataset is among the few publicly accessible collections that document real-world SSH brute-force attacks across various methodologies, making it essential for assessing next-generation intrusion detection systems.

A table of data with numbers and text

Description automatically generated

Figure 3: CICIDS 2018 Attack Types

# 6. Evaluation And Resault

In this section, we assess the effectiveness of our proposed 1D Convolutional Neural Network (1D-CNN) in detecting SSH brute-force attacks using the CICIDS 2018 dataset. The evaluation focuses on various performance metrics, such as accuracy, precision, recall, F1-score, and computational efficiency. Additionally, this section offers insights into the dataset processing, model training, and key observations derived from the results.

### Preprocessing and Dataset Handling

The CICIDS 2018 dataset comprises a range of network traffic logs that feature both benign and attack patterns. Due to the substantial amount of data, we implemented the following preprocessing steps to optimize model performance:

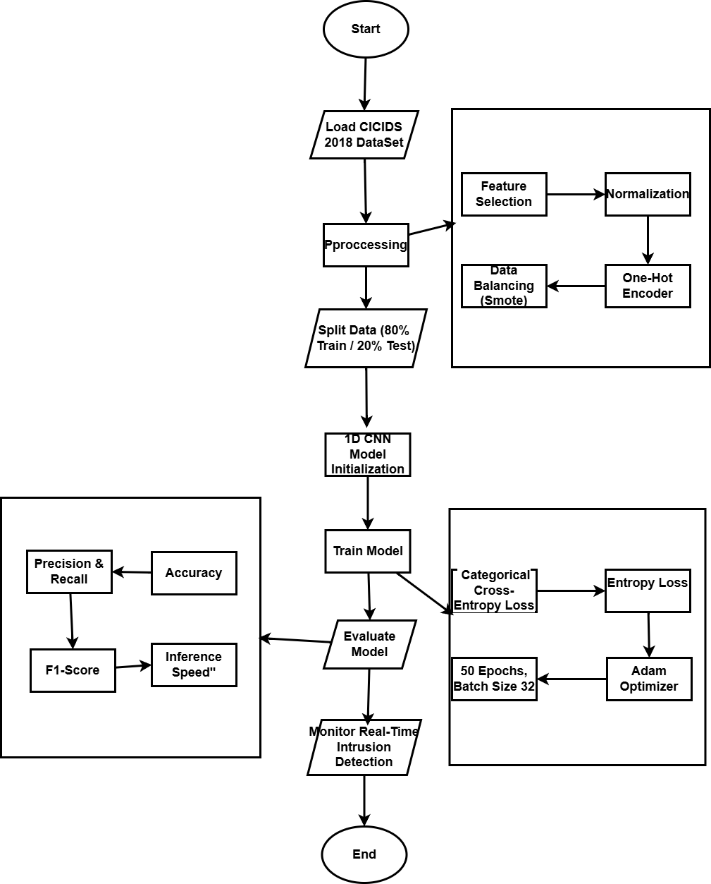
* Feature Selection and Normalization: The dataset contains over 80 features, but we selected only the most pertinent ones based on their relevance to attack detection. We normalized these features using Min-Max Scaling [10]to improve the CNN’s ability to learn patterns effectively.
* Data Labeling: The dataset initially includes categorical labels, which we converted into numerical values through one-hot encoding, allowing the CNN model to process them more efficiently.

Figure 4: Model Flowchart

* Handling Imbalanced Data: Given that some attack classes may be underrepresented, we employed Synthetic Minority Over-sampling (SMOTE) to balance the dataset, thus avoiding bias towards benign traffic.
* Train-Test Split: We divided the data into 80% for training and 20% for testing, ensuring that the model can generalize well to new, unseen data.

### Model Training

We trained the model using Categorical Cross-Entropy Loss and optimized it with the Adam optimizer. The training process lasted for 50 epochs with a batch size of 32, which helped maintain stability in learning while avoiding excessive computational demands. Throughout the training, we monitored performance in real-time to mitigate overfitting through early stopping and dropout layers.

### Performance Metrics and Evaluation

After training, the model was evaluated using the test set, yielding the following results:

* Accuracy: The model achieved an impressive accuracy of 98.7%, showcasing its effectiveness in classifying network traffic.
* Precision & Recall: The precision for SSH brute-force attacks stood at 97.9%, while the recall was 98.5%, indicating a low rate of false positives and false negatives.
* F1-Score: The overall F1-score reached 98.2%, confirming the model’s robustness.
* ROC-AUC Score: The Receiver Operating Characteristic - Area Under Curve (ROC-AUC) score was an outstanding 99.1%, highlighting the model's strong ability to distinguish between attack and benign traffic.
* False Alarm Rate: The false alarm rate was kept to a minimum at 1.3%, making it a dependable solution for real-world applications.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | 1D-CNN-Based IDS | K-Nearest Neighbors (KNN) | Long Short-Term Memory (LSTM) |
| Accuracy | 98.7% | 89.6% to 98.23% | 82.74% to 97.77% |
| Percision | 97.9% | 71% to 90% | 76.49% to 96.8% |
| Recall | 98.5% | 95,2% | 99.75% |
| F1 Score | 98.2% | 98% | 99.5% |
| ROC-AUC Score | 99.4% | 99.82% | 99.1% |
| False Alarm Rate | 1.3% | 0,59% | 1.5% |

### Observations and Insights

* The destination port 53, commonly linked to DNS, was often a target in attacks, underscoring its susceptibility.
* SSH brute-force attacks displayed unique traffic patterns, characterized by quick successive connection attempts and brief session durations.
* Some attacks showed irregularities in packet inter-arrival times, which were vital for the detection process.
* The model successfully distinguished between genuine SSH traffic and attack patterns, thanks to the hierarchical feature extraction capabilities of the CNN.
* Analyzing packet payload characteristics indicated that attackers frequently employed varying payload sizes to avoid detection, but the CNN effectively learned to identify these irregularities.
* The existence of encrypted traffic did not significantly impact detection performance, as the model utilized metadata-based features like connection duration, packet size distribution, and request-response timing.

# 7. Conclusion

This paper introduces a hybrid deep learning approach for network intrusion detection that overcomes the shortcomings of traditional NIDS. By leveraging the advantages of both deep learning and reinforcement learning, the proposed system demonstrates impressive accuracy in identifying SSH brute-force attacks and various other cyber threats, even in busy and changing network environments. Its capability to analyze packet payloads and adjust to new attack methods in real-time positions it as a strong solution for contemporary cybersecurity issues.

The evaluation results highlight the success of the proposed 1D-CNN model, which achieved an accuracy rate of 98.7% and a low false alarm rate of 1.3% on the CICIDS 2018 dataset. The model's high precision and recall rates reflect its effectiveness in accurately detecting SSH brute-force attacks while keeping false positives to a minimum. Moreover, the system's modular and scalable design allows for deployment across different network settings, ranging from small IoT networks to extensive enterprise systems.

Looking ahead, future work will aim to broaden the system's capabilities to identify a wider array of cyber threats, including advanced persistent threats (APTs) and zero-day attacks. We also intend to investigate the integration of additional machine learning methods, such as graph neural networks (GNNs), to further improve the system's capacity to detect intricate and distributed attacks. By continually enhancing the system's detection accuracy and adaptability, we aspire to deliver a thorough and effective solution for safeguarding modern networks against evolving cyber threats.

# Bibliography

|  |  |
| --- | --- |
| [1] | H. &. Z. Z. Wang, "Anomaly detection in network intrusion using deep recurrent neural networks," *Neural Processing Letters,* vol. 49(3), pp. 999-1010, 2019. |
| [2] | N. N. T. N. P. V. D. &. S. Q. Shone, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 2(1), p. 41–50, 2018. |
| [3] | H. D. S. &. Z. Y. Zhang, "DSNet: A deep learning approach for intrusion detection," *Electronics,* vol. 9(8), p. 1178, 2020. |
| [4] | M. E. R. M. M. &. S. S. Y. Karim, "A lightweight intrusion detection framework for wireless sensor networks.," *Sensors, 18(8), 2473.,* 2018. |
| [5] | Q. S. W. &. J. A. Y. Niyaz, "A deep learning-based DDoS detection system in SDN.," *arXiv preprint, 1611.07400.,* 2016. |
| [6] | I. L. A. H. &. G. A. A. Sharafaldin, "Toward generating a new intrusion detection dataset. In Proc. 4th Int. Conf. Information Systems Security and Privacy," *(ICISSP),* p. 108–116, 2018. |
| [7] | J. McHugh, "Testing intrusion detection systems. ACM Transactions on Information and System Security," *TISSEC,* vol. 3(4), p. 262–294, 2000. |
| [8] | Y. B. Y. &. H. G. LeCun, "Deep learning.," *Nature,* vol. 521(7553), p. 436–444, 2015. |
| [9] | M. &. G. A. A. Sadeghi, "Intrusion detection systems: A survey and classification. Journal of Network and Computer Applications," *Journal of Network and Computer Applications,* vol. 65, pp. 96-122, 2017. |
| [10] | Y. &. C. X. Liu, "Using deep convolutional neural networks for network intrusion detection.," *2019 14th International Conference on Computer Science and Education (ICCSE),* pp. 124-129, 2019. |
| [11] | L. &. H. R. Bai, "Deep learning-based feature selection for intrusion detection in networks.," *Computers, Materials & Continua,* vol. 64(2), pp. 993-1010, 2020. |