International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-13 Issue-5, January 2025

A Robust Framework for Detecting Brute-Force Attacks through Deep Learning Techniques

Nouf Awadh, Hawazen Zaid, Samah Al-ajmani

Abstract: ***A considerable concern arises with the precise identification of brute-force threats within a networked environment. It emphasizes the need for new methods, as existing ones often lead to many false alarms, as well as delays in real-time threat detection. To tackle these issues, this study proposes a novel intrusion detection framework that utilizes deep learning models for more accurate and efficient detection of brute-force attacks. The framework's structure includes data collection and preprocessing components performed at the outset of the study using the CSE-CICIDS2018 dataset. The design architecture includes data collection and preprocessing steps. Feature extraction and selection techniques are employed to optimize data for model training. Further, after building the model, various attributes are extracted from the data from feature selection to be used in the training. Then, the construction of multiple architectures of deep learning algorithms, which include Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) models. Evaluation results show CNN and LSTM achieved the highest accuracy of 99.995% and 99.99% respectively. It showcases its ability to detect complex attack patterns in network traffic. It indicates that the CNN network got the best optimum results with a test time of 9.94 seconds. This establishes CNN as an effective method, achieving high accuracy quickly. In comparison, we have surpassed the accuracy of current methods while addressing their weaknesses. The findings are consistent with the effectiveness of CNN in brute-force attack detection frameworks as a more accurate and faster alternative, increasing the capability of detecting intrusions on a network in real-time.***

Keywords: Deep Learning, Brute Force Attack, IDS, CSE-CICIDS2018

I. INTRODUCTION

Recent advancements in technologies such as artificial intelligence (AI), big data, and the Internet of Things (IoT) have significantly increased our support for the Internet [1]. However, this increased connectivity also increases anomalous behaviors, making cyber-threat protection one of the most pressing challenges in information technology today [2]. The proliferation of tiny, connected devices capable of transmitting personal data to the internet further intensifies this issue, exacerbating the ongoing battle between security systems and cyber-criminals. Brute force attacks are a significant threat to network security [3].

Unlike other sophisticated tactics, these attacks rely on sheer computational power and trial-and-error algorithms to crack encrypted data like passwords or Data Encryption Standard keys [4]. What makes brute force attacks especially dangerous is that they can closely resemble legitimate network traffic, making it challenging for organizations relying on perimeter-based security measures to defend against them effectively.

Intrusion detection systems (IDS), firewalls, and antivirus software are critical components of cybersecurity defense [5]. IDS plays a crucial role as an early warning system against network assaults. It gathers a large volume of malicious attack data, compares behavioral patterns against a database of known threats, and detects intrusions based on this comparison [6]. IDS is invaluable in combating known threats and evolving attacks, such as new ransomware variants [7].

Deep learning, a subset of artificial intelligence, introduces a powerful tool for enhancing IDS capabilities [8]. Unlike traditional machine learning, deep learning utilizes multi-layered neural networks that can autonomously learn and process features, adapting to the characteristics of complex data sets [9]. As the volume of data grows rapidly, deep learning ability to automatically process and extract critical features makes it a highly effective approach for analyzing large datasets. Designing optimal deep learning architectures and balancing layers enables the system to identify key patterns and make informed decisions about network security threats [10].

The main goal of this study is to propose a deep learning-based intrusion detection framework for identifying brute-force attacks. We will assess the accuracy of deep learning models in detecting brute force attack patterns using the CSE-CICIDS2018 dataset [11]. Then, we will evaluate the effectiveness of various deep learning architectures in differentiating between normal and malicious network traffic. Finally, we will compare the performance of deep learning-based intrusion detection systems with traditional security methods in detecting and mitigating brute force attacks [12].

- Propose a deep learning-based intrusion detection framework for detecting brute force attacks in network traffic.
- Measure the effectiveness of deep learning algorithms in accurately identifying brute force attack patterns using the CSE-CICIDS2018 dataset.

Manuscript received on 31 October 2024 | First Revised Manuscript received on 10 December 2024 | Second Revised Manuscript received on 17 December 2024 | Manuscript Accepted on 15 January 2025 | Manuscript published on 30 January 2025.

*****Correspondence Author(s)

Nouf Awadh*, College of Computers and Information Technology, Taif University, Taif, SA. Email ID: S44580346@students.tu.edu.sa, ORCID ID: 0009-0008-4396-2812

Hawazen Zaid, College of Computers and Information Technology, Taif University, Taif, SA. Email ID: S44580271@students.tu.edu.sa, ORCID ID: 0009-0009-9773-7394

Dr. Samah Al-ajmani, Department of Information Technology, College of Computer and Information Technology, Taif University, Taif, SA. Email ID: s.ajmani@tu.edu.sa, ORCID ID: 0009-0000-7152-9559

- Evaluate various deep learning architectures' ability to differentiate brute force attack anomalies from normal network traffic.
- Compare the performance of proposed deep learning-based intrusion detection systems with conventional security approaches in detecting and mitigating brute force attacks.

The paper organization is as follows: The sections below provide a short overview of IDS and deep learning, the second section discusses related work, the third section discusses research methodology, the fourth section discusses the proposed model, the fifth section discusses the analysis of the requirement, the sixth section discusses the results, and the seventh section discusses the conclusions and future works.

II. THE RELATED WORKS

This section discusses the developments in Intrusion Detection Systems (IDS) as a cornerstone of network defense and device security. We will evaluate the limitations of traditional intrusion detection methods in addressing modern cyber threats, especially brute force attacks. The review will demonstrate the need to evolve current IDS methods further to address the ongoing challenges on networks. It will highlight how deep learning offers a better solution to these limitations.

Provided a comprehensive overview of the latest developments in AI-powered cybersecurity. They highlighted how technologies such as machine learning (ML), deep learning (DL), natural language processing (NLP), and reinforcement learning (RL) are transforming cybersecurity measures. The study examined the utilization of these technologies for threat detection, network security, data protection, and cyber threat responses, outlining the growing reliance on AI to counter increasingly complex cyber threats. The research identified major topics in AI-driven cybersecurity and cluster analysis to organize these topics into subcategories, offering a structured view of the field. The findings suggest that AI will continue to play a pivotal role in enhancing cybersecurity frameworks and inspiring innovation in various areas of cyber protection [13].

Also systematically reviewed the current trends, challenges, and innovations in cybersecurity in 2024. The review focused on the evolving nature of cyber threats, driven by increased digitalization and the growing need for adaptive security measures. The study examined the global trends and current conditions in the cybersecurity landscape, presenting future strategies for combating emerging threats. Additionally, it emphasized the integration of AI and ML for automating cyber threat detection and response systems. The study highlighted the necessity of collaboration between stakeholders and continuous technological adoption to address the dynamic nature of the cyber threat environment [2].

Delved into the advancements in user authentication and authorization in cybersecurity in 2024. The study explored the application of Machine Learning (ML) and Deep Learning (DL) models for improving authentication and authorization mechanisms to safeguard personal information and digital assets. The research offered a detailed analysis of the datasets, pre-processing techniques, and ML/DL algorithms commonly employed in this area. Furthermore, it examined state-of-the-art methods and challenges, discussing future research directions. This systematic review serves as a key resource for researchers by offering insights into ML/DL-based user authentication and authorization systems and their role in bolstering cybersecurity defenses [14].

A. Intrusion Detection Systems in Cybersecurity

Cybersecurity holds significant importance in today's society [15]. The instantaneous connectivity to the global network exposes individuals and organizations to cyber threats. An Intrusion Detection System (IDS) is designed to identify and respond to potential threats or malicious activities [16]. Although considerable efforts have been made to enhance the security and privacy of computer systems, these issues persist, with no system in the world being entirely impervious to security breaches.

Intrusion Detection Systems (IDS) play a critical role in modern cybersecurity by identifying potential security breaches and anomalous activities within network infrastructures. There are two main types of IDS: signature-based and anomaly-based. Signature-based systems rely on predefined attack patterns and may struggle with novel or evolving threats. Anomaly-based systems, on the other hand, can struggle with high false-positive rates due to the complexity of distinguishing between legitimate and malicious network behavior.

As cyber-attacks become more complex, there is an increasing demand for advanced Intrusion Detection System (IDS) solutions. Recent discussions have focused on integrating artificial intelligence (AI), particularly deep learning, into IDS to address these limitations. Deep learning can autonomously learn from data to detect subtle and evolving patterns, making it well-suited for environments with high data variability. It has been suggested that deep learning-based IDS significantly improves the detection of complex threats, such as brute-force attacks.

Moreover, deep learning can reduce the operational limitation of IDS by automating feature learning and anomaly detection, making IDS systems more scalable and efficient. However, there are challenges associated with deep learning in IDS, such as the need for vast datasets, high computational costs, and the risk of overfitting specific attack patterns. To address these challenges, hybrid models that combine traditional IDS methods with deep learning architectures have been proposed to balance computational efficiency with detection accuracy.

Furthermore, various types of network attacks exist [17]. An IDS serves as a defense at the network level to safeguard computer networks. Intrusions or threats manifest as anomalies within a network. Advanced non-traditional intrusion detection systems are typically used to identify known and unknown attacks on a network from internal and external sources. However, many of the techniques used in modern intrusion detection systems struggle to effectively deal with the dynamic and complex nature of cyber-attacks on computer networks due to hackers' continuous attempts. With the continuous evolution of malicious threats, existing network security systems have proven insufficient to protect computer systems.

The intrusion detection system can take the form of a host-based IDS (HIDS) or a network-based IDS (NIDS). Network administrators tend to employ host-based intrusion detection systems to monitor and analyze activities on specific machines. HIDS often offers the advantage of accessing encrypted information transmitted over a network. However, its management can be challenging as each host requires configuration and information management.

Presented the IntruDTree model, a machine learning-based security model that prioritizes security features and builds a tree-based intrusion detection model based on these important features. Their model not only accurately predicts unseen test cases but also reduces the computational complexity by reducing the dimensionality of the features. The effectiveness of the model was evaluated through some experiments on cybersecurity data using a set of evaluation criteria. The results were compared with traditional machine learning methods [18].

Proposed an intrusion detection system (IDS) for the automotive sector. The system uses a two-step algorithm to detect potential cyber-attacks. It filters all messages on the controller area network (CAN-Bus) using spatial and temporal analysis. If a set of messages is flagged as potentially malicious, it is analyzed by a Bayesian network, which calculates the probability of classifying a particular event as an attack. A pilot campaign was conducted to evaluate the efficiency and effectiveness of the method using classical evaluation parameters such as test accuracy. The results were compared to a dataset based on cyber-attack. The initial experimental results obtained in the test scenario are promising. They demonstrated that the method performs well in detecting common cyber-attacks such as DDoS, Fuzzy, and Impersonating, achieving a good score [19].

Analyzed learning characteristics using representative features. Their work demonstrated that network intrusions occurring outside the scope of the learned data in the feature space can evade detection by ML-NIDS (Machine Learning-based Network Intrusion Detection Systems). Furthermore, they designed an active session to be classified before it extends beyond the ML-NIDS training dataset's detection range, effectively preventing the bypassing of the system. Various experiments confirmed that this method could detect intrusion sessions early enhancing the robustness of existing ML-NIDS [20].

Various machine learning techniques are used in cybersecurity, but there is still much work to be done in intrusion detection. Recently, the concept of hybrid learning has gained attention from information security specialists in the fight against cyber threats. In their work [21], proposed a hybrid method of swarm intelligence and evolution for feature selection, known as PSO-GA. This method was applied to the CICIDS-2017 dataset before training the model.

Proposed an effective network intrusion detection system based on machine learning and feature selection techniques. They examined the performance of four machine learning techniques: Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT) systems for intrusion detection. In addition, feature selection techniques were employed to select important features. Among the techniques used, the RF technique achieved the best performance, outperforming other techniques, with an accuracy of 99.72%. Table 2.1 summarizes the key research trends on Intrusion Detection Systems (IDS) in cybersecurity [22].

B. Deep Learning in Cybersecurity

In this section, we will focus on the theoretical foundation of deep learning for detecting attacks and threats in cybersecurity. The analysis will compare deep learning with traditional machine learning models, emphasizing the autonomy and adaptability of neural networks in extracting features and learning from them to continuously identify new threat patterns. The section will evaluate how deep learning architectures reshape detection capabilities, particularly in recognizing brute force attack patterns in large datasets.

Rapidly developing image recognition, natural language processing, bioinformatics, and speech recognition applications have led to the widespread use of Deep Learning (DL) algorithms [9]. These algorithms play an important role in solving complex problems and offer several advantages over traditional Machine Learning (ML) techniques. DL involves multi-layered ML algorithms that excel at learning high-level abstractions from large-scale data. By automatically learning feature representations using multiple non-linear hidden layers, DL eliminates the need for manual feature engineering [23]. It means that companies in the cybersecurity industry can update their intrusion detection systems (IDSs) and malware detection systems at a minimal cost without the need to hire engineers to re-engineer features as new types of malware or network attacks emerge [24].

Conducted a study on developing an intrusion detection system (IDS) using deep neural networks (DNNs) to identify and categorize unexpected cyberattacks. Their research involved a comprehensive assessment of deep neural networks and traditional machine learning classifiers across publicly available malware datasets. The primary innovation was the introduction of a scalable hybrid DNN framework, Scale-Hybrid-IDS-AlertNet, created to monitor real-time network traffic and host-level events. Their findings are particularly significant in the current research on intrusion detection systems within network-level security, as they tackle the challenge of sophisticated attacks by leveraging deep learning for flexible, high-dimensional feature extraction [25].

Proposed using an attention-based CNN-LSTM model to enhance intrusion detection in Industry 4.0 environments utilizing the IDS 2018 dataset. The incorporation of the attention mechanism was aimed at boosting the effectiveness of current deep learning techniques in IDS. The study applied advanced deep learning models for network security, specifically focusing on tackling the challenges presented by extensive datasets and continuously evolving threats [26].

Developed a comprehensive deep intrusion detection model that combines a Stacked Denoising Autoencoder with an Extreme Learning Machine (SDAE-ELM) and a Deep Belief Network with Softmax (DBN-Softmax) for improved network and host intrusion detection. The study addressed the extended training times and low classification accuracy typically associated with existing

deep learning models. Their experiments, which included multiple datasets such as KDD Cup99 and UNSW-NB15, showed that their integrated models outperformed traditional machine-learning approaches. It emphasizes the effectiveness of innovative deep learning architectures in enhancing intrusion detection performance [27].

Introduced a deep learning-based intrusion detection system tailored for Internet of Things (IoT) environments. Their system aimed to tackle the security challenges brought about by the growing number of connected devices. The model integrated the Spider Monkey Optimization (SMO) algorithm for optimal feature selection and a Stacked-Deep Polynomial Network (SDPN) to classify normal and anomalous behavior effectively. Through extensive analysis, they demonstrated that their system outperformed existing IDSs regarding accuracy, precision, recall, and F-score. The study emphasizes the need for advanced intrusion detection techniques, emphasizing the importance of effective feature learning and dataset management [28].

Developed an advanced network intrusion detection system (NIDS) using deep learning techniques to combat the increasing vulnerabilities in network security. The study involved training the model on real-time traffic data to improve attack detection capabilities. The results suggested that deep learning methods effectively address the evolving landscape of cyber threats. It aligned with the current emphasis on intrusion detection approaches, showcasing the potential of deep learning to enhance detection accuracy and adapt to new attack vectors [29].

Discussed the challenges of identifying malicious traffic and zero-day attacks in the context of the growing number of IoT devices. They proposed a hybrid technique that combined Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to optimize network parameters for intrusion detection, resulting in an accuracy of 98.73% with a low False Positive Rate. This study adds to the current discourse on enhancing deep learning approaches for Intrusion Detection Systems (IDS) and underscores the importance of effective feature learning [30].

Highlighted the significance of advanced security solutions in light of evolving network threats. They developed a hybrid Deep Learning-Based Network Intrusion Detection System (HDLNIDS) utilizing convolutional and recurrent neural networks to improve detection accuracy, achieving an average accuracy of 98.90%. Their findings further support ongoing research efforts in optimizing IDS through deep learning methodologies, demonstrating the potential of hybrid models for real-world applications [31].

Focused on the challenges of intrusion detection in the dynamic landscape of cyber threats. They proposed a two-stage hybrid model that integrated Long-Short-Term Memory (LSTM) and Auto-Encoders (AE) to enhance attack detection. Their research contributes to a more comprehensive understanding of deep learning's role in IDS development. It highlights the need for innovative methods to keep pace with the rapid evolution of attack techniques [32].

Used deep learning models to tackle the increasing challenges of cyber-attacks on Critical Cyber-Physical Systems (CPSs), particularly in Industrial Control Systems (ICS) and Internet of Things (IoT) networks. The interconnected nature of these systems generates a large volume of network traffic, making it challenging for traditional machine-learning methods to detect anomalies efficiently. To address these challenges, the researchers proposed a two-stage methodology using Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) models, along with a Deep Sparse Autoencoder (DSAE) for feature extraction. The approach was evaluated using the IoT-23 and LITNET-2020 datasets, and the results demonstrated significant improvements in anomaly detection compared to state-of-the-art techniques. This method offers a robust solution for securing critical CPSs against cyber threats [33].

Focused on enhancing cybersecurity in Industry 4.0-based wireless sensor networks (WSNs), vulnerable to cyberattacks due to their continuous data collection and real-time communication. The study introduced a predictive framework for prioritizing and preventing cyber intrusions in Industry 4.0 WSNs. The framework integrated machine learning and deep learning models, including Decision Tree, Multilayer Perceptron (MLP), and Autoencoder, to detect and classify cybersecurity threats. The Decision Tree and MLP models were utilized for multidimensional classification, achieving accuracy rates of 99.48% and 99.52%, respectively. The Autoencoder model, applied for binary classification, demonstrated an accuracy of 91%. Benchmark models such as Random Forest (RF) and Logistic Regression (LR) were also implemented for performance comparison, with the proposed framework surpassing these benchmarks. This research reinforces cybersecurity measures in Industry 4.0 WSNs by providing a robust, multi-criteria intrusion detection and prevention system [34].

Highlighted the growing threats posed by network intrusions to data privacy, necessitating effective detection methodologies. They proposed a deep learning-based intrusion detection system that employs a chaotic optimization strategy for improved classification accuracy to address this issue. The model utilized data cleansing, M-squared normalization, and the Extended Synthetic Sampling approach to balance unbalanced datasets, followed by feature extraction using kernel-assisted principal component analysis. The final attack classification was performed using a Gated Attention Dual Long Short-Term Memory (Dugat-LSTM) model, achieving impressive accuracy rates of 98.76% on the TON-IOT dataset and 99.65% on the NSL-KDD dataset. The study emphasizes the importance of robust feature selection and optimization in intrusion detection. It highlights the necessity of utilizing advanced deep learning techniques to enhance detection accuracy against evolving cyber threats [35].

C. Comparative Analysis of Detection Approaches

This section will provide a thorough analysis of various detection methods in the cybersecurity domain, focusing on their effectiveness in identifying different types of continuous attacks and threats. We will compare traditional machine learning techniques, such as decision trees and support vector machines, with various advanced deep learning architectures, highlighting their strengths and weaknesses. Key aspects of the comparison will include

the accuracy, scalability, and adaptability of the algorithms to evolving threats.

We will also evaluate the performance of ensemble methods and hybrid methods that combine machine learning and deep learning, examining how these methods can enhance detection rates and reduce false positives. By assessing the effectiveness of each approach in real-world scenarios, we aim to identify the most suitable techniques for different cybersecurity challenges, ultimately contributing to the development of more robust intrusion detection systems.

Conducted a comprehensive review of anomaly-based intrusion detection systems (AIDS) using machine learning (ML) and deep learning (DL) techniques. The study addressed shortcomings in previous research, such as the use of outdated datasets and shallow analysis, and applied 10 ML algorithms to evaluate performance across a recent CICIDS 2017 dataset. They found that the k-NN, Decision Tree, and Naive Bayes algorithms best detected web attacks. They contribute to the current research by emphasizing the importance of benchmarking various ML algorithms for effective intrusion detection in real-world environments [36].

Focused on deep learning architectures for creating adaptive and resilient network intrusion detection systems capable of detecting both known and zero-day attacks. Using the UNSW-NB15 dataset, the research demonstrated how deep neural networks could adapt to growing threats, strengthening the possibility of flexible IDS solutions. The study is related to the current work by emphasizing deep learning's role in detecting emerging and sophisticated cyber threats in modern networks [37].

Introduced the MapReduce-Based Intelligent Model for Intrusion Detection (MR-IMID), which applies machine learning to enhance intrusion detection in large-scale IoT networks. Their work addressed the challenge of analyzing vast data volumes and achieved high detection accuracy during both training (97.7%) and validation (95.7%) phases. Their contribution underscored the importance of accuracy in detecting a wide range of cyber-attacks in real-time [38].

Investigated network intrusion detection by utilizing a Kippo honeypot to classify SSH attacks through machine learning. Their study implemented four classification algorithms—Decision Tree, Naive Bayes, SVM, and Random Forest—demonstrating that Decision Tree and Random Forest achieved an accuracy and F1 score of 0.92. Their work highlights the effectiveness of honeypot data in enhancing intrusion detection systems, particularly for SSH attacks [39].

Proposed an Intelligent Intrusion Detection System (IDS) specifically for Vehicular Ad hoc Networks (VANETs). The system employed a hybrid approach that integrates an Adaptive Neuro-Fuzzy Inference System (ANFIS) and Convolutional Neural Networks (CNN) to detect both known and unknown attacks. The system showed impressive performance metrics, with accuracies reaching up to 99.2% for PortScan attacks. This research addressed the need for robust IDS frameworks capable of handling a variety of attack vectors in dynamic environments [40].

Developed five machine learning classifiers using the CSE-CIC-IDS2018 dataset, which captures a wide range of modern network attacks. This dataset provides a much-needed update to previous datasets like KDD Cup'99, which are outdated in light of evolving cyber threats. Ilyas's findings emphasize the importance of utilizing contemporary data for the development of effective intrusion detection systems, which resonates with the current research's focus on adapting methodologies to new attack patterns [41].

Developed an enhanced anomaly-based intrusion detection model (EIDM) tailored to classify 15 types of network traffic behaviors, including 14 attack types, with 95% accuracy using the CICIDS 2017 dataset. Their work compared EIDM to state-of-the-art deep learning IDS models, demonstrating superior classification accuracy. This research supports the current study by showcasing the importance of deep learning in multi-class classification for IoT environments, particularly in enhancing detection abilities [42].

Proposed a novel stacking ensemble model, DIS-IoT, which integrates multiple deep learning models for IoT intrusion detection. Tested on datasets such as ToN_IoT, CICIDS2017, and SWaT, DIS-IoT excelled in multi-class classification with low false positive rates, outperforming traditional models. This work is relevant to the present research by highlighting the effectiveness of ensemble methods for complex IoT traffic classification and detection of diverse attacks [43].

Introduced the CNN-Focal model to address the limitations of traditional intrusion detection systems (IDS) in handling complex network traffic. This model employs threshold convolution and SoftMax multi-class classification to improve the accuracy and efficiency of detecting abnormal traffic. It performed well on publicly available datasets, demonstrating its applicability in real-world environments. This research offers a novel approach to enhancing deep learning-based IDS models, focusing on improving classification efficiency in dynamic network environments [44].

Proposed deep learning (DL) models for IoT intrusion detection using two datasets, NSL-KDD and UNSW-NB15. The study used a filter-based feature selection approach and compared the performance of a deep neural network (DNN) and convolutional neural network (CNN). Both models achieved high accuracy rates, and explainable AI techniques like LIME and SHAP were added to improve interpretability and model transparency. It aligns with the current research by exploring deep learning methods for intrusion detection and emphasizing the importance of model interpretability in securing IoT networks [45].

Introduced a deep learning-based IDS using a chaotic optimization strategy and feature extraction techniques like kernel-assisted PCA, achieving accuracy rates of 98.76% on the TON-IOT dataset and 99.65% on the NSL-KD dataset. Their work underscored the significance of advanced deep learning methods in intrusion detection, reinforcing the necessity for robust feature selection and optimization to combat evolving cyber threats. It emphasized innovative approaches to improve detection accuracy in the face of increasing attacks [35].

Explored the challenges of processing and analyzing log files generated from cloud computing environments, focusing on the detection of brute force attacks. By using machine learning algorithms such

as Naive Bayes, Decision Tree, Random Forest, and Support Vector Machine, the research demonstrated that Decision Trees and Random Forests were particularly effective, achieving 87% accuracy in detecting malicious traffic within access logs. This study contributes to enhancing anomaly detection systems in cloud computing by showing how traditional ML models can be adapted to process large volumes of log data [46].

Developed a machine learning-based intrusion detection system (IDS) designed for smart grid infrastructure, specifically targeting man-in-the-middle (MITM) attacks in advanced metering infrastructure (AMI). The research used real-world hardware setups and communication protocols (Modbus TCP/IP, MQTT) to emulate attacks in a real-time testbed environment. The results indicated that the Random Forest algorithm outperformed others in detecting these intrusions, providing a practical solution for securing smart grid communications. This study highlights the importance of using real-time data in IDS for critical infrastructure like smart grids [47].

Focused on brute force attacks targeting secured shell (SSH) servers. Over 20 days, the research logged over one million brute force attacks across five servers. The study emphasizes the growing need for cyber forensics as a defense mechanism against such attacks, stressing that cyber-crimes are not only growing in frequency but also in severity. The findings support the idea that brute force attacks are a key vector for more significant cybercrimes, highlighting the need for stronger intrusion detection and prevention mechanisms [48].

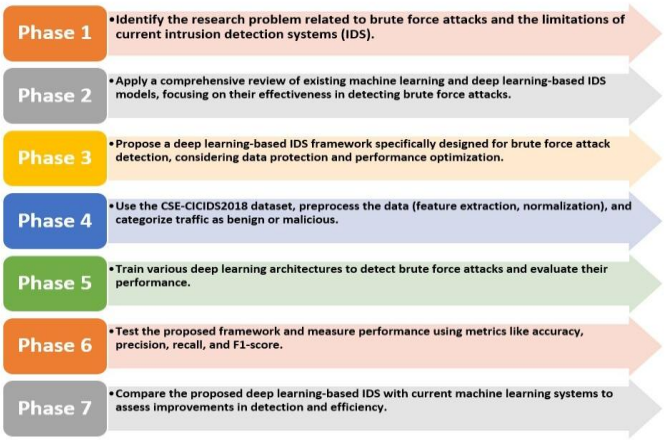
Created the Server-based Network Attack (SNA) dataset to study the real-time detection of network attacks in server-based environments. They used advanced AI techniques and introduced a meta-RF-GNB (MRG) model, which combines Random Forest and Gaussian Naive Bayes. This model accurately detected network attacks such as UDP, SYN, and HTTP floods [49].

Focused on improving intrusion detection systems (IDS) for the Internet of Things (IoT) by introducing an attention-based convolutional neural network (ACNN). The model combines the Inception architecture with a spatial attention mechanism to enhance feature extraction and localization. Tested on the CIC-IDS 2018 dataset, the ACNN achieved an accuracy of 98.11% in detecting and categorizing 14 types of attacks. Its compact size of 21.172 MB and computational efficiency makes it suitable for deployment in resource-constrained IoT devices, contributing to advancements in IoT security by improving detection accuracy with minimal computational overhead [50].

Proposed a hybrid intrusion detection model that combines convolutional neural networks (CNN) and bidirectional long-short-term memory (BiLSTM) networks, which is enhanced with an attention mechanism. The research utilized the UNSW-UB15 and CSE-IC-IDS2018 datasets, applying the NCR-SMOTE resampling technique and recursive feature elimination for feature selection. Their approach achieved a classification accuracy of 98.3% on the CSE-IC-IDS2018 dataset. This study advances the field of network intrusion detection by combining deep learning architectures with attention mechanisms to improve prediction accuracy in complex network environments [51].

III. RESEARCH METHODOLOGY

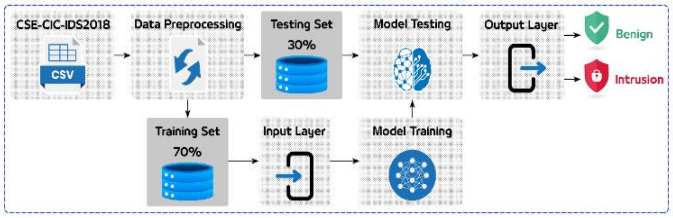
The previous section literature review included a thorough theoretical analysis, focusing on various essential characteristics of the current machine and deep learning-based intrusion detection systems (IDS). This research covered their effectiveness in detecting brute force attacks, data protection strategies, and the performance of different network architectures. The study will also investigate models to enhance deep learning-based intrusion detection to overcome brute force attacks in cybersecurity networks. Additionally, experiments will be conducted to evaluate and validate the proposed system's performance using the CSE-CICIDS2018 benchmark dataset to assess its ability to detect brute force attacks effectively. The research methodology will be structured into several phases, as the following figure:



[Fig.1: Research Methodology]

IV. THE PROPOSED MODEL

The proposed system is designed to enhance intrusion detection capabilities by applying advanced deep learning techniques. This section provides an overview of the system's structure, highlighting its core features and functional flow as presented in Fig. 2. Emphasis is placed on integrating data preprocessing, feature selection, and implementing multiple deep learning models. The system aims to detect brute force attacks efficiently by analyzing network traffic patterns and anomalies. Each phase of the process is systematically outlined to demonstrate how the framework handles the complexities of modern cybersecurity threats.

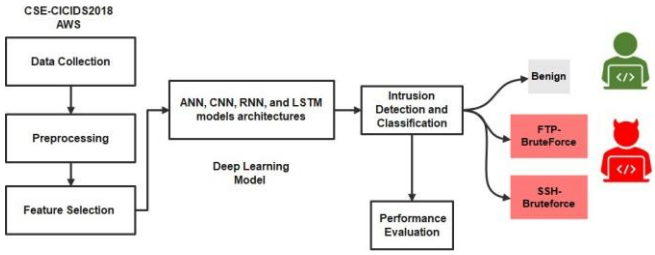


[Fig.2: Model Overview] [52]

A. System Architecture

In this section, the system architecture of the proposed

intrusion detection framework is presented, describing its essential components and data flow. The architecture is structured to handle large-scale network traffic data. It begins with data collection and preprocessing from the CSE-CICIDS2018 dataset. Following this, feature extraction and selection are applied to optimize the data for model input. The design of various deep learning models is then outlined. It highlights their role in identifying anomalies and detecting brute force attacks within the processed data. Each component is carefully integrated to ensure system efficiency and accuracy.



[Fig.3: The Proposed Model Architecture]

B. CSE-CICIDS2018 Data Collection and Preprocessing

In this section, the CSE-CICIDS2018 dataset is employed for data collection, serving as the basis for the proposed intrusion detection framework scenario. It is generated systematically, which involves creating user profiles that represent various network behaviors and attack scenarios. The dataset is designed to simulate real-world network conditions. It incorporates both legitimate traffic and multiple types of attacks, including brute force, DoS, DDoS, and web attacks. It provides comprehensive coverage of network activities, allowing for robust testing and evaluation of deep learning-based detection models.

Preprocessing is applied to the raw data to ensure its suitability for model training. It includes filtering irrelevant features, handling missing values, and transforming network traffic data into structured formats. Feature extraction is performed using CICFlowMeter-V3, which generates 80 features (sample of presented features in Table- I) from the captured network traffic, ensuring that the most relevant characteristics of both normal and malicious traffic are represented. It ensures that the dataset is optimized for effective analysis and model training, enabling the detection of anomalies and brute force attack patterns with high accuracy.

Table 1: Sample of the CSE-CICIDS2018 Dataset Features

Feature	Description
fl_dur	Flow duration
tot_fw_pk	Total packets in the forward direction
tot_bw_pk	Total packets in the backward direction
tot_l_fw_pkt	Total size of packets in the forward direction
fw_pkt_l_max	Maximum size of packets in the forward direction
fw_pkt_l_min	Minimum size of packets in the forward direction
fw_pkt_l_avg	The average size of packets in the forward direction
fw_pkt_l_std	The standard deviation of the size of packets in the forward direction

Normalization is then applied to standardize the extracted features, ensuring consistency in the data across varying scales. The proposed phase mitigates the impact of feature value dissimilarities and improves the model's learning efficiency.

Once normalization is completed, the traffic is categorized as benign or malicious based on known attack signatures and behavioral patterns. The categorization composes the basis for training the intrusion detection models, enabling them to differentiate between legitimate network activity and potential threats effectively. To apply normalization in the CSE-CICIDS2018 dataset, the Min-Max Normalization equation is commonly used, which is presented as the next equation (4.1):

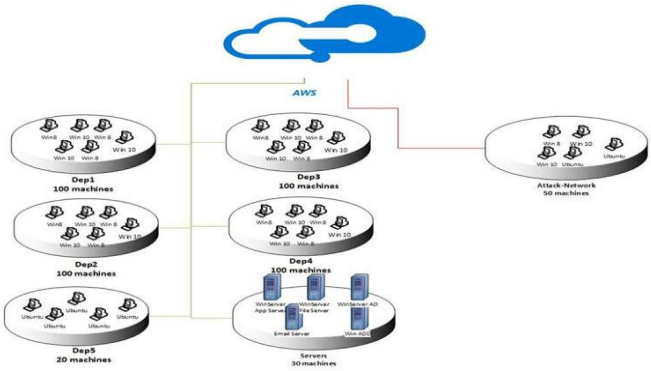
$$Xnorm = \frac{X - Xmin}{Xmax - Xmin} \quad \ldots \quad (4.1)$$

Where **X** represents the original value of the CSE-CICIDS2018 dataset.

- $\mathbf{X_{min}}$ Represents the minimum value of each specific feature of the CSE-CICIDS2018 dataset.
- $\mathbf{X_{max}}$ Represents the maximum value of each specific feature of the CSE-CICIDS2018 dataset.
- $\mathbf{X_{norm}}$ Represents the normalized value of each specific feature of the CSE-CICIDS2018 dataset.

C. Feature Extraction and Selection

Feature extraction and selection are essential phases in applying an effective intrusion detection system (IDS) by refining raw network traffic data into meaningful insights. During feature extraction, important characteristics are identified from the raw data that capture both normal and malicious network behaviors. Tools such as CICFlowMeter-V3 are used to extract a wide range of features. It includes flow duration, packet size, total bytes, and packet inter-arrival times.



[Fig.4: Topology of the CSE-CICIDS2018 Network Data] [16]

The proposed features from the CSE-CICIDS2018 dataset encapsulate the structure and timing of network traffic, which are essential for identifying patterns related to brute force attacks, denial of service (DoS) attacks, and other security threats as presented in Fig. 4 The extracted features serve as the basis for the following analysis and classification phases [53].

Feature extraction is essential for modeling as the dataset contains numerous features. However, it has many of which may be unessential, redundant, or insignificant for detecting specific attacks. Therefore, feature selection is performed to identify and keep only the most useful features, which enhances the model's performance by reducing complexity. Common techniques for feature

selection include statistical methods such as correlation analysis, and machine learning-based approaches such as Principal Component Analysis (PCA). These techniques assess the significance of each feature by measuring its contribution to accurate predictions and its relationship with other features, allowing irrelevant attributes to be discarded.

A feature selection process improves computational efficiency by reducing the dataset's dimensionality and mitigating the risk of overfitting. It can occur when models are trained on noisy or irrelevant data. Focusing on the most relevant features makes the model more robust and better generalizes unseen data.

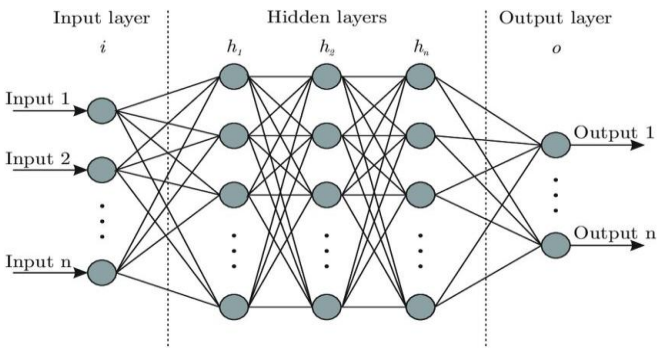
Brute force attack detection features such as login attempts, request frequency, and packet size variability may be prioritized, as they are directly associated with attack patterns. It ensures that the model is well-equipped to differentiate between benign network traffic and malicious activities based on a refined feature set.

Additionally, the feature selection enhances the model's interpretability. Understanding which specific features contribute to detecting anomalies and intrusions becomes easier. The complexity of the model is reduced, allowing it to focus on key indicators of abnormal network behavior, improving both the detection rate and accuracy. The integration of feature extraction and selection phases ensures the model is efficient. It can be more scalable and capable of detecting complex attack patterns such as brute force attempts.

D. Deep Learning Model Design

This section discusses deep learning models for intrusion detection, focusing on selecting architectures that effectively identify anomalies in network traffic. Several deep learning models, such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks, are considered for their ability to process complex, high-dimensional data.

Each model is designed to manage different aspects of intrusion detection, with CNNs used for pattern recognition in spatial data, RNNs and LSTMs applied to temporal data for sequence analysis, and ANNs serving as general-purpose models for classification tasks. The selection of the suitable model architecture is driven by the dataset characteristics and the specific network behaviors being analyzed.



[Fig.5: Common Artificial Neural Network Modeling Architecture] [18]

The models are designed with multiple layers, including input, hidden, and output layers, each fine-tuned using hyper-parameter optimization to enhance performance as shown in Fig. 5. Regularization techniques such as dropout and batch normalization are applied to prevent overfitting, while activation functions like ReLU is used to introduce non-linearity into the models, allowing them to capture more complex patterns.

The design process also includes strategies for balancing computational efficiency with detection accuracy, ensuring that the models can operate in real-time network environments. Overall, the model design emphasizes the ability to adjust to diverse attack patterns, including brute force, DoS, and web-based intrusions, ensuring robust detection capabilities.

E. Modeling Training and Implementation

This section discusses the training and implementation of deep learning models, focusing on optimizing their ability to detect brute force attacks and additional network intrusions. The CSE-CICIDS2018 dataset is used to train the models, with the data divided into training and testing sets to evaluate performance.

During the training phase, the models are exposed to both benign and malicious network traffic, allowing them to learn the distinguishing features of each. Supervised learning techniques are applied, where the models adjust their weights and biases based on the error between predicted and actual outcomes. The iterative process ensures the models improve their accuracy in classifying network traffic.

Implementing the trained models involves fine-tuning hyper-parameters such as learning rates, batch sizes, and the number of epochs to achieve optimal performance. Various performance metrics, including accuracy, precision, recall, and F1-score, are used to evaluate the models' effectiveness.

Cross-validation is also employed to prevent overfitting and ensure that the models generalize well to unseen data. Once the models are trained, they can be easily deployed as a real-time intrusion detection system, continuously monitoring network traffic, identifying anomalies, and flagging potential attacks. The implementation we aim for ensures the models perform well on historical data and can detect new and growing threats in dynamic network environments.

i. Artificial Neural Network (ANN) Model

In this section, the Artificial Neural Network (ANN) model is utilized to detect network traffic anomalies. The ANN is designed with multiple interconnected layers, including an input layer, hidden layers, and an output layer, which work together with the backpropagation process to classify traffic as benign or malicious.

The model is trained using supervised learning, where labeled data from the CSE-CICIDS2018 dataset display the learning process. Each node in the hidden layers applies an activation function, such as ReLU, to introduce non-linearity, allowing the model to capture complex patterns from the proposed data. During the training phase, the ANN modifies the weights between nodes through backpropagation, minimizing the error between predicted and actual outputs. Optimization techniques, such as stochastic gradient descent (SGD), provide efficient

learning and convergence. Regularization methods, including dropout, are applied to prevent overfitting, thereby improving the model's generalization to unseen data. Once trained, the ANN is evaluated based on key performance metrics such as accuracy, precision, and recall, ensuring it effectively differentiates between normal and malicious network traffic in real-world applications.

ii. Convolutional Neural Network (CNN) Model

This section applies the Convolutional Neural Network (CNN) model to intrusion detection tasks, using its ability to automatically extract features from high-dimensional network traffic data. The CNN architecture consists of multiple layers, including convolutional, pooling, and fully connected layers, designed to capture spatial patterns within the input data.

The main advantage of CNNs lies in their capability to identify local and hierarchical features through convolutional layers, making them highly effective for detecting various patterns in network traffic. Pooling layers further enhance efficiency by reducing the dimensionality of the data while preserving essential information, which is particularly beneficial in handling large-scale datasets.

The CNN is trained using labeled data from the CSE-CICIDS2018 dataset, where the model learns to recognize patterns associated with benign and malicious traffic. Activation functions, such as ReLU, introduce non-linearity, enabling the model to identify complex relationships in the data that more straightforward models might miss.

Additionally, CNNs require minimal feature engineering due to their automatic feature extraction capabilities, reducing the need for manual intervention. Backpropagation is employed to update the model's parameters, and optimization techniques like Adam or SGD are used to minimize the error. Once trained, the CNN is evaluated using metrics such as accuracy, precision, and recall, ensuring its robust performance in detecting anomalies and brute force attacks in real-world network environments.

iii. Recurrent Neural Network (RNN) Model

In this section, the Recurrent Neural Network (RNN) model is applied to intrusion detection, particularly focusing on its strength in handling sequential data. RNNs are designed with feedback connections that allow information to persist across time steps, making them highly effective for analyzing temporal patterns in network traffic.

The feature is especially valuable for detecting brute force and other time-dependent attacks, where the sequence of actions or packet flows is critical for identifying malicious behavior. The model captures the relationship between consecutive network events, enabling more accurate detection of anomalies that unfold over time.

The RNN is trained using the CSE-CICIDS2018 dataset, with network traffic data fed into the model as sequences. During training, the RNN updates its internal state based on past inputs and learning patterns in benign and malicious traffic over time. The model's parameters are optimized using backpropagation through time, which adjusts weights to minimize prediction errors. Techniques such as gradient clipping are employed to address vanishing gradient problems, ensuring stable training.

Once trained, the RNN is evaluated using performance metrics like accuracy, recall, and F1-score to assess its effectiveness in detecting time-based network attacks, making it a valuable tool for identifying complex, evolving intrusion patterns.

iv. Long Short-Term Memory (LSTM) Model

The Long Short-Term Memory (LSTM) model is utilized in this section for intrusion detection, focusing on its ability to handle long-term dependencies in sequential data. LSTM, an advanced variant of Recurrent Neural Networks (RNN), is particularly suited for detecting attacks that unfold over extended periods, such as brute force or slow, stealthy network intrusions. The model overcomes the limitations of traditional RNNs by incorporating memory cells, gates (input, forget, and output), and a mechanism that selectively retains important information from earlier time steps while discarding irrelevant data.

The LSTM model is trained using sequential data from the CSE-CICIDS2018 dataset, allowing it to capture short-term and long-term network traffic patterns. Using backpropagation through time (BPTT), the model adjusts its weights and biases to minimize errors while learning patterns in benign and malicious sequences.

Gradient vanishing, a common issue in standard RNNs, is mitigated in LSTMs, ensuring stable learning across long sequences. Once trained, the LSTM is evaluated using accuracy, precision, and recall metrics. It ensures distinguishes between normal traffic and sophisticated attack patterns effectively. The ability to maintain context over a longer duration makes LSTM particularly valuable for identifying complex, evolving intrusion patterns in real-world network environments.

F. Intrusion Detection and Attack Classification

In this section, the main focus is on intrusion detection and attack classification within the proposed framework. Deep learning models, including ANN, CNN, RNN, and LSTM, classify network traffic as benign or malicious based on patterns and anomalies found in the data.

The primary objective is to accurately identify various types of network intrusions, such as brute force attacks, denial of service (DoS), and other forms of cyberattacks. Each model utilizes specific features and characteristics in the dataset to enhance detection accuracy and minimize false positives.

The process of attack classification involves analyzing extracted features and assigning labels based on learned patterns from the training data. The models can effectively distinguish between normal network behavior and malicious activities by identifying deviations from typical traffic patterns. Once classified, the system directly flags any detected threats, enabling swift and effective responses.

The performance of each model is assessed using accuracy, precision, recall, and F1-score. It ensures their ability to detect known and unknown attack types consistently. The proposal guarantees the framework can adapt to evolving cyber threats. It provides a strong protection tool from unknown threats in dynamic network environments.

G. System Performance Evaluation

In this section, we evaluate system performance using standard metrics. Accuracy and precision are vital to the assessment, while recall and F1-score are essential in measuring detection significance. We test each ANN, CNN, RNN, and LSTM model on the CSE-CICIDS2018 dataset. We analyze various attack types for classification accuracy, including brute force and DoS. The models' ability to differentiate between benign and malicious traffic is discussed in detail. These evaluations highlight the different strengths and drawbacks of each model. Additionally, time performance is a vital aspect of the analysis.

H. Accuracy and Detection Metrics

Accuracy measures the proportion of correctly classified instances out of the total instances. It can be calculated in the next equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total}}$$

Precision quantifies the ratio of correctly predicted positive observations to the total predicted positives. It can be calculated in the next equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall measures the ratio of correctly predicted positives to all actual positives. It can be calculated in the next equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F1-Score provides a harmonic mean of precision and recall, balancing the two. It can be calculated in the next equation:

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives.
- Time Performance Analysis

Execution time for training and testing the model calculating model performance to compare detection times for proposed models.

V. REQUIREMENTS ANALYSIS

This section analyzes hardware and software requirements to provide the proposed model's successful implementation. Hardware specifications are based on the need for efficient data processing. It includes model training, including memory, CPU, and GPU resources. The system must sustain large-scale data handling due to the complexity of the CSE-CICIDS2018 dataset. Tools for data preprocessing, model training, and evaluation are needed on the software side. Deep learning frameworks, including TensorFlow and Keras, are recommended for model development. The operating system should be compatible with these tools to ensure seamless integration for optimal performance.

A. Hardware Requirements

This section outlines the hardware requirements necessary for implementing deep learning models. Google Colaboratory is used due to its accessible cloud-based environment, which offers significant computational power. Colab provides CPUs, GPUs, and TPUs to facilitate intensive machine learning tasks [54]. The availability of such resources makes it highly suitable for complex operations including training deep neural networks.

GPUs, especially the NVIDIA Tesla K80 with 12GB of VRAM, are utilized in Colab for parallel processing and accelerating model training. The GPU's architecture is optimized for handling large-scale matrix operations, a core component in training deep learning models. It makes the GPU highly effective when working with large datasets, significantly improving model training time and efficiency.

TPUs are another available resource in Colab, explicitly developed by Google for machine learning tasks. These custom-built hardware accelerators further optimize model performance. TPUs allow for faster computations compared to traditional GPUs, particularly in tasks requiring high levels of matrix multiplication, such as those involved in training neural networks. GPUs and TPUs sustain the challenging requirements of large-scale data analysis and deep learning tasks.

B. Software Requirements

This section outlines the essential software features for modeling development. TensorFlow is selected due to its flexibility in building and training deep learning models. As a high-level API, Keras simplifies model configuration and improves TensorFlow's usability. These tools are compatible with cloud-based environments like Google Colaboratory [54]. It ensures seamless integration and execution. As the primary programming language, Python is widely supported by both TensorFlow and Keras [16]. It makes the ideal choice for machine learning tasks.

The operating system should efficiently support these frameworks. Windows operating system is often preferred because of its compatibility. With deep learning libraries and ease of use in machine learning environments. Pre-installed libraries such as NumPy, Pandas, and Matplotlib are essential for data manipulation and visualization. These libraries facilitate data preprocessing, a vital step in preparing the dataset for model training. Integrating these tools is necessary for an efficient workflow, from data preprocessing to model evaluation.

Virtual environments managed through Python are recommended to isolate dependencies and avoid software conflicts. It ensures that different versions of TensorFlow, Keras, and other libraries. It also improves the system's overall stability during the development process. Utilizing these software tools makes the system robust, adaptable, and well-suited for large-scale deep learning tasks.

VI. RESULTS

The significant findings and outcomes of the research work are demonstrated in a pictorial form in this chapter. The efforts made to compute performance measures for the models employed within the dataset are analyzed in detail. The dataset is treated with a great deal of focus on features that will be relevant for classification purposes, and the importance of data cleansing and normalization processes is considered. Convolutional Neural Network (CNN),

Artificial Neural Network (ANN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) model performance metrics were compared concerning accuracy, precision, recall, and F1-score. Each algorithm demonstrates varying degrees of effectiveness in identifying and classifying different types of attacks. Outliers of the selected features were underscored since univariate methods such as Feature Selection using the SelectKBest approach were used for model build-up increase. There are, however, significant improvements in the detection rates achieved in the area of deep learning models that is CNN and LSTM as first draggable which are better than the rest in the matters of attack classification.

A. Pre-Processing

In this section, the focus is on providing the data preprocessing stage which is done to develop a ready dataset to be used during model training. In order to maintain the quality of the datasets, incomplete data entries are either removed or replaced with some meaningful values. It is important to identify and eliminate duplicates before model training in order to avoid introducing any biases. The numerical variables have been subjected to standard scaling techniques, which facilitates achieving a common ground for all variables. Features in the categorical data were transformed as well, with label encoding and one-hot encoding techniques being deployed to allow the features to work in machine learning models. The timestamp feature was treated as unnecessary in classification so it was removed from the dataset to prevent replication. Finally, the SelectKBest procedure was implemented to ensure that only features that are relevant to the classification task were used in the input of the models.

B. Feature Selection

In this section, feature engineering should be viewed within the context of improving the performance of machine learning models by lessening the amount of data while still keeping the most relevant predictive variables. The SelectKBest method employs the ANOVA F-test to identify 10 of the most informative features for the classification task from the dataset that initially comprised 78 numerical and two categorical attributes.

Among the characteristics selected include 'fwd_pkt_len_max' and 'bwd_pkt_len_max' that help identify the maximum forward and backward packet length necessary for the communication pattern detect irregularities. 'flow_pkts_s' and 'fwd_pkts_s' assess the quantity of packets in a flow and those moving towards the forward direction per unit time, which detects changes in traffic. Similarly, 'bwd_pkts_s' captures the rates at which the packets in the backward direction are transmitted as well and can depict anomalies of activity.

An example of a feature that enables an analyst to forecast attacks aimed at modifying packet size is 'pkt_len_max', which is the maximum packet size in a network flow. 'psh_flag_cnt', for example, is also a good attack pattern indicator as it counts the number of PUSH flags present in the traffic. The features' init_fwd_win_byts' and 'fwd_act_data_pkts' provide insight into 'flow control mechanisms during attacks' as they denote the initial window size and the total number of forward active data packets, respectively.

'fwd_seg_size_min' is also important because it captures the smallest size of the fwd. Segment, which is usually changed during malicious activities.

The selected attributes are very significant in addressing the detection and classification of attacks which then improves the effectiveness and the prediction power of the model implemented to higher levels.

C. Model Training Settings

In this section, the training settings for each model, namely, CNN, ANN, RNN, and LSTM, will be described to point out their configurations and execution times respectively. For the Artificial Neural Network (ANN) as shown in Table- II, a three-layer architecture is designed using 64, 32, and 16 units in each of the respective layers having the ReLU activation function. The Multi-Class Classification is performed using softmax function in the output layer and the optimizer used is the Adam optimizer at a 0.001 learning rate. The training spans ten epochs with batch size 256, and the training execution is time-bound to evaluate model efficiency.

Table 2: ANN Training Settings

Layer Type	Units	Activation Function
Input Layer	64	ReLU
Hidden Layer 1	32	ReLU
Hidden Layer 2	16	ReLU
Output Layer	3	Softmax
Optimizer	Adam (learning rate 0.001)	
Loss Function	Categorical Crossentropy	
Epochs	10	
Batch Size	256	

As for The Long-Short-Term Memory (LSTM) model as shown in Table 6.2, a three-layer-lattice LSTM architecture is used with a structure of 64 32 16 units with dropout used as a regularizer.

LSTMs are trained on reshaped 3D input data due to their temporal nature. Categorical cross-entropy is taken as the loss function. Likewise, the model was trained for ten epochs, using a batch size of 256, and again,
training time was recorded.

Table 3: LSTM Training Settings

Layer Type	Units	Activation Function
LSTM Layer 1	64	ReLU
Dropout Layer 1	-	-
LSTM Layer 2	32	ReLU
Dropout Layer 2	-	-
LSTM Layer 3	16	ReLU
Dropout Layer 3	-	-
Output Layer	3	Softmax
Other Settings	Value	
Optimizer	Adam (learning rate 0.001)	
Loss Function	Categorical Crossentropy	
Epochs	10	
Batch Size	256	

The same configuration prevalent in an LSTM model is followed in the Recurrent Neural Network (RNN) as shown in Table- III, RNN layers take the place of the LSTM layers.

The training remains nearly the same as that of the LSTM model. However, the configuration is modified with dropout overpowering and the Adam optimizer to address temporal dependencies present in the input data.

Table 4: RNN Training Settings

Layer Type	Units	Activation Function
RNN Layer 1	64	ReLU
Dropout Layer 1	-	-
RNN Layer 2	32	ReLU
Dropout Layer 2	-	-
RNN Layer 3	16	ReLU
Dropout Layer 3	-	-
Output Layer	3	Softmax
Other Settings	Value	
Optimizer	Adam (learning rate 0.001)	
Loss Function	Categorical Crossentropy	
Epochs	10	
Batch Size	256	

Finally, the Convolutional Neural Network (CNN) model has been built which consists of two 1D convolutional layers that have 64 filters and 32 filters each followed by fully connected and max pooling layers. The input data as shown in Table- IV, is then modified into 3-D arrays in order to allow integration of the Conv1D layers since categorical crossentropy is used as the loss function. Like in the other models, the training of the CNN is executed for the duration of ten epochs for the purpose of measuring the execution time in order to consider the performance efficiency.

Training parameters such as the number of layers and units, activation functions, optimizers, and batch sizes are all selected with a bias toward classification accuracy but without subjecting the model to excessive computational cost.

Table 5: CNN Training Settings

Layer Type	Filters/Units	Kernel Size
Conv1D Layer 1	64	3
MaxPooling1D Layer 1	-	2
Conv1D Layer 2	32	3
MaxPooling1D Layer 2	-	2
Flatten Layer	-	-
Dense Layer	16	-
Output Layer	3	-
Other Settings	Value	
Optimizer	Adam (learning rate 0.001)	
Loss Function	Categorical Crossentropy	
Epochs	10	
Batch Size	256	

D. Confusion Matrix Analysis

The results of the confusion matrix are examined in order to determine the classification performance for different types of attacks according to the models.

All models show a high true negative rate for Benign, indicating a good performance in correctly identifying benign traffic. The false positives for Benign also remain low, meaning that there is a good degree of accuracy in distinguishing between normal behavior and attacks.

There is also a success rate for true positive counts for FTP-BruteForce as it mainly denotes the model's capabilities in detecting denial-of-service attacks.

It is also especially interesting to note that false negatives never occur in FTP-BruteForce; this further emphasizes the effectiveness of the detection methods used.

Table 6: Summary of the Confusion Matrices for the ANN, LSTM, RNN, and CNN Models

Metric	ANN	LSTM	RNN	CNN
True Negative (Benign)	132451	132444	132416	132450
False Positive (Benign)	0	3	6	0
False Negative (Benign)	1	5	30	2
True Positive (FTP-BruteForce)	7832	7832	7832	7832
False Positive (FTP-BruteForce)	0	0	0	0
False Negative (FTP-BruteForce)	0	0	0	0
True Positive (SSH-Bruteforce)	23497	23533	23538	23535
False Positive (SSH-Bruteforce)	43	7	2	5
False Negative (SSH-Bruteforce)	2	2	2	2

While Class SSH-Bruteforce in which fabrication attacks are represented, all the true positive values are highly significant across all models. However, there is a small increase in false negatives, particularly in the case of the RNN model.

It can also be attributed to a weakness of the model. This evidence seems to indicate that while fabrication attacks were detected entirely, some measures were required to attain better accuracy. The false positive rates for SSH-Bruteforce attacks have been low suggesting confidence in the models predicting any actual attacks.

Analysis of the confusion matrix indicates that the models are effective in the classification of types of attacks. The approaches applied have been verified as effective with the presence of high true positive rates and low rates of both false negatives and false positives.

E. Attack Detection Results

The rate of recovery and detection of various kinds of attacks and their classification was achieved by developing several models which are summarized in Table 6.6 and Fig. 6. All the models achieve good figures for the evaluation metrics, namely accuracy, precision, recall, and F1 score, which show the reliability of these models in detecting anomalies. CNN model outshines them all, as it attains an accuracy of 99.995% on classifying the traffic into normal and attack traffic suggesting the capability of the model. This model also manages to post similarly high values of precision, recall, and the F1 score giving a picture of all metrics giving the same results.

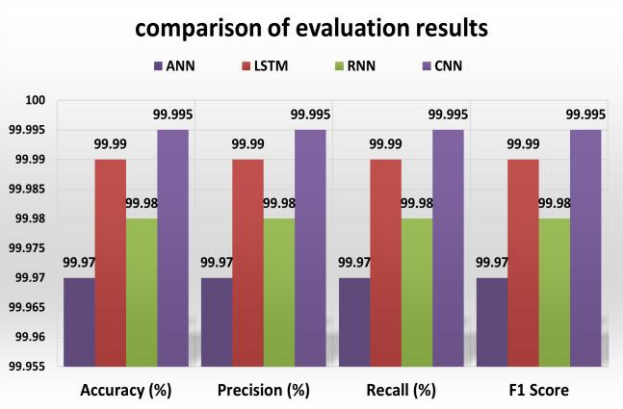
Next comes the LSTM model that comes second attaining 99.990% accuracy making it closely follow CNN Model. The precision and recall are also relatively high making it quite effective in attack identification and not misclassifying normal traffic. There is a small difference in performance between the LSTM and CNN models which makes it quite interesting. It implies that approaches built around

recurrent neural networks are successful for the task of time sequence data analysis focusing towards anomaly detection in this case. Like all models represented in Table- VII, these also do not show significant deviation from the values that were expected.

Table 7: Comparison of Evaluation Results for the ANN, LSTM, RNN, and CNN Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score
ANN	99.97	99.97	99.97	99.97
LSTM	99.990	99.990	99.990	99.990
RNN	99.98	99.98	99.98	99.98
CNN	99.995	99.995	99.995	99.995

Only slightly lower results are gained by the RNN model, which achieves an accuracy of 99.98%. It is, however, still quite effective but slightly lags behind the performance offered by LSTM and CNN models with regard to precision and recall. This observation suggests that the RNN, while proficient at attack detection, can be improved in terms of its capability to classify different types of attacks.



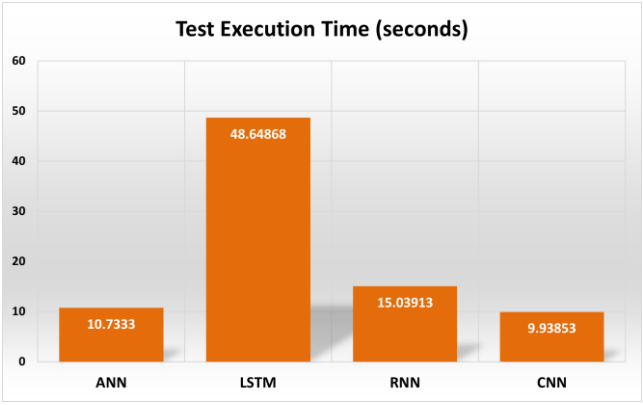
[Fig.6: Comparison of Evaluation Results for the ANN, LSTM, RNN, and CNN Models]

On the contrary, the ANN model has been rated as the least performance-wise, holding all others constant. Though it still attained a remarkable 99.97% accuracy. The precision values and recall ratios are all in line with the overall accuracy, which indicates an even detection across different attack types. It comes under expectations that the ANN model will be less competent than others for complex attack schemes.

The models successfully detect attacks, with the CNN and LSTM getting the best results.

F. System Performance Metrics

In Fig. 7 and Table- VIII, the time taken to complete all the test runs for each model serves as a measure for assessing their efficiency. The CNN model comes out as the most time-effective, performing the task in just 9.93853 seconds. It goes on to show that the CNN model has the most advantages of high accuracy and economical use of computing resources, thus making it very appropriate for real-time applications.



[Fig.7: Summary of the Test Execution Time Results]

Table 8: Summary of the Test Execution Time Results

Model	Test Execution Time (Seconds)
ANN	10.7333
LSTM	48.64868
RNN	15.03913
CNN	9.93853

With a relatively low-test execution time of 10.7333 seconds, the ANN model does not take a lot of time to make predictions and hence can be classified as fast in terms of artificial neural network approaches. It is also accurate in its predictions and optimizes the time taken; however, it is slower in comparison to the CNN model in practice but within a reasonable range.

In contrast, the RNN model's total computation takes quite more extended than the previous model, with a test execution time averaging at 15.03913 seconds. It implies that a recurrent neural network can perform tasks but is more tedious in training. As a result, this longer time can be caused by the predominant linear structure of the RNN, which involves more steps in tracking the dependencies spanning a sequence of time. Regardless of the CNN and ANN models taking less time, this model also captures a good amount of merit in terms of time and precision.

For the LSTM model, the time taken to make the predictions is the longest. The stern comparison establishes a time frame of 48.64868 seconds for making predictions based on a set of parameters denoted as the test execution time. This hypothesis leads to the finding that the longitudinal routing architecture decouples the temporal signal from the processing, making it accurate but costly. Considering the positives, this model can be useful in high-accuracy situations but may not always work in settings with negligible processing times.

The system performance metrics show that CNN and ANN models are the best in execution time, as presented in Table-VII. LSTM and RNN provide excellent accuracy, but their higher performance cost indicates a lower efficiency.

G. Discussion

As shown in Table -IX: Similar works of attack detection models using the CSE-CIC-IDS-2018 dataset. The study by [55], used the GPC-FOS method which is a combination of Gaussian Process Classification and FOS, to detect attacks against the CSE-CIC-IDS-2018 dataset. This method achieves 97.22% accuracy. These

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

results have shown that using GPC-FOS is effective, but compared to other methods, it presents an opportunity for improvement.

Table 9: Similar Works of Attack Detection Models using the CSE-CIC-IDS-2018 Dataset

Reference	Methods	Results (%)
[55]	GPC-FOS	97.22
[56]	SVM	94.5
[57]	MLP and LSTM	95
[58]	LSTM + AM	96.97
[59]	CatBoost	92.41
[60]	Random Tree	98.3
Ours	CNN	99.995

While used Support Vector Machines (SVM) on the same dataset and managed a 94.5% detection rate. While the SVM method is robust, its efficiency is a little lower than recent algorithms, which implies that the classical approaches center on all the dynamics associated with the contemporary attack domains [56].

Combined MLP and Long Short Term Memory networks to record 95% detection accuracy. It advocates for a good combination of machine learning and deep learning advanced techniques. However, the incremental advantage of the latter over the former is minimal, implying good performance versus complexity [57].

According to integrating attention mechanisms with LSTM models enhanced detection capability to 96.97%, which is better than older models. This improvement demonstrates the worth of attention to relevant time-series features of the network traffic, ensuring better detection rates and relatively lower computation costs [58].

Described a method of predicting network traffic using the CatBoost algorithm, which achieved a 92.41% accuracy result. Although this method is quite useful, its performance is generally low compared to some methods in the same class, and this might be due to its dependence on dataset features or imbalanced datasets [59].

According to the Random Tree method attained an accuracy of 98.3%. It shows the significance of decision tree-based techniques on anomaly detection in the modern world, especially when employed with large and complex datasets like the CSE-CIC-IDS-2018 international dataset [60].

On the other hand, our Convolutional Neural Network (CNN)-based model performed with an accuracy rate of 99.995%. CNN's power of capturing spatial hierarchies of datasets enables the model to accurately classify normal and abnormal traffic with very low errors, increasing its capabilities.

VII. CONCLUSION

The proposal investigates and showcases the capability of deep learning models in formulating a robust intrusion detection framework. It is exclusively aimed at the detection of brute force attacks. Using the CSE-CIC-IDS 2018 dataset, different models were tested, most of which were based on Artificial Neural Networks such as ANNs, RNNs, and LSTMs as well as CNNs. Among these, CNN had the highest validation accuracy of 99.995%, thus demonstrating its potential in identifying complicated attack patterns in the flow of network traffic. The performance of LSTM and RNN models also yielded favorable results as the effectiveness of recurrent architecture to recognize time-related aspects within the data failed to disappoint. However, while ANN recorded a good performance, it was lower in comparison with CNN in terms of precision, recall, and general detection ability.

The feature selection stage of the exercise, during which the SelectKBest was incorporated, led to better model performance. Predictive accuracy was also maintained high by reducing the dataset's multiplicity factor by identifying the critical features such as 'fwd_pkt_len_max' and 'bwd_pkt_len_max.' Such features are integral in the detection of brute force attack anomalies and thus support the significance of a packet-level analysis for intrusion detection systems.

The developed framework makes an important step in cyberspace security, especially in brute-force attack detection. The results of this study are promising and provide a strong basis for future work directed toward developing intrusion detection systems based on machine learning with better scalability, adaptability, and efficiency. Advancements in algorithm development and hybrid integration are essential to counter increasingly sophisticated cyber threats.

Further work should address more efficient architectures and optimization strategies. As CNN has reported outstanding results, building models that include, for instance, transformers could improve detection abilities when it comes to more heterogeneous and dynamic network traffic. Randomly extending the scope of the dataset towards a greater diversity of attack types as well as their more significant numbers, on the other hand, could also strengthen the models' robustness and generalization. Also, deploying these models in real space would allow for transfer learning as an avenue for rapid adjustment to new attack angles, improving the framework's utility.

DECLARATION STATEMENT

Authors are required to include a declaration of accountability in the article, counting review-type articles, that stipulates the involvement of each author. The level of detail differs; Some subjects yield articles that consist of isolated efforts that are easily voiced in detail, while other areas function as group efforts at all stages. It should be after the conclusion and before the references.

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been sponsored or funded by any organization or agency. The independence of this research is a crucial factor in affirming its impartiality, as it has been conducted without any external sway.
- **Ethical Approval and Consent to Participate:** The data provided in this article is exempt from the requirement for ethical approval or participant consent.
- **Data Access Statement and Material**

Availability: The adequate resources of this article are publicly accessible.

- **Authors Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. J. Li, M. S. Herdem, J. Nathwani, and J. Z. Wen, 'Methods and applications for Artificial Intelligence, Big Data, Internet of Things, and Blockchain in smart energy management', *Energy AI*, vol. 11, p. 100208, 2023. DOI: https://doi.org/10.1016/j.egyai.2022.100208
2. W. S. Admass, Y. Y. Munaye, and A. Diro, 'Cyber security: State of the art, challenges, and future directions', Cyber Secur. Appl., p. 100031, 2023. DOI: https://doi.org/10.1016/j.csa.2023.100031
3. M. F. K. Shah, M. Md-Arshad, A. A. Samad, and F. A. Ghaleb, 'Comparing ftp and ssh password brute force attack detection using k-nearest neighbor (k-nn) and decision tree in cloud computing', Int. J. Innov. Comput., vol. 13, no. 1, pp. 29–35, 2023. DOI: https://doi.org/10.11113/ijic.v13n1.386
4. P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, 'Internet of things: Security and solutions survey', Sensors, vol. 22, no. 19, p. 7433, 2022. DOI: https://doi.org/10.3390/s22197433
5. S. Kumar, S. Gupta, and S. Arora, 'Research trends in network-based intrusion detection systems: A review', Ieee Access, vol. 9, pp. 157761–157779, 2021. DOI: https://doi.org/10.1109/ACCESS.2021.3129775
6. Z. Azam, M. M. Islam, and M. N. Huda, 'Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree', IEEE Access, 2023, Accessed: Sep. 29, 2024. DOI: https://doi.org/10.1109/ACCESS.2023.3296444
7. Z. A. E. Houda, B. Brik, and L. Khoukhi, '"Why Should I Trust Your IDS?": An Explainable Deep Learning Framework for Intrusion Detection Systems in the Internet of Things Networks', IEEE Open J. Commun. Soc., vol. 3, pp. 1164–1176, 2022, DOI: https://doi.org/10.1109/OJCOMS.2022.3188750
8. 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', Trans. Emerg. Telecommun. Technol., vol. 32, no. 1, p. e4150, Jan. 2021, DOI: https://doi.org/10.1002/ett.4150
9. T. Srinivas, G. Aditya Sai, and R. Mahalaxmi, 'A comprehensive survey of techniques, applications, and challenges in deep learning: A revolution in machine learning', Int. J. Mech. Eng., vol. 7, no. 5, pp. 286–296, 2022. DOI: https://doi.org/10.70593/978-81-981367-4-9_2
10. M. Z. Gunduz and R. Das, 'Cyber-security on smart grid: Threats and potential solutions', Comput. Netw., vol. 169, p. 107094, 2020. DOI: https://doi.org/10.1016/j.comnet.2019.107094
11. A. A. Hagar and B. W. Gawali, 'Deep Learning for Improving Attack Detection System Using CSE-CICIDS2018', NeuroQuantology, vol. 20, no. 6, p. 3064, 2022. https://www.researchgate.net/publication/362619753_Deep_Learning_for_Improving_Attack_Detection_System_Using_CSE-CICIDS2018
12. J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, 'A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions', Electronics, vol. 9, no. 7, p. 1177, Jul. 2020. DOI: https://doi.org/10.3390/electronics9071177
13. M. Paramesha, N. L. Rane, and J. Rane, 'Artificial Intelligence, Machine Learning, and Deep Learning for Cybersecurity Solutions: A Review of Emerging Technologies and Applications', Partn. Univers. Multidiscip. Res. J., vol. 1, no. 2, Art. no. 2, Jul. 2024, DOI: https://doi.org/10.5281/zenodo.12827076
14. Z. T. Pritee, M. H. Anik, S. B. Alam, J. R. Jim, M. M. Kabir, and M. F. Mridha, 'Machine learning and deep learning for user authentication and authorization in cybersecurity: A state-of-the-art review', Comput. Secur., vol. 140, p. 103747, May 2024, DOI: https://doi.org/10.1016/j.cose.2024.103747
15. M. Abdel-Rahman, 'Advanced Cybersecurity Measures in IT Service Operations and Their Crucial Role in Safeguarding Enterprise Data in a Connected World', Eig. Rev. Sci. Technol., vol. 7, no. 1, pp. 138–158, 2023. DOI: https://doi.org/10.3390/electronics9071177
16. A. Nagisetty and G. P. Gupta, 'Framework for detection of malicious activities in IoT networks using keras deep learning library', in 2019 3rd international conference on computing methodologies and communication (ICCMC), IEEE, 2019, pp. 633–637. DOI: https://doi.org/10.1109/ICCMC.2019.8819688
17. S. M. Albladi and G. R. Weir, 'User characteristics that influence judgment of social engineering attacks in social networks', Hum.-Centric Comput. Inf. Sci., vol. 8, no. 1, pp. 1–24, 2018. DOI: https://doi.org/10.1177/0093650215627483
18. I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, 'IntruDTree: A Machine Learning Based Cyber Security Intrusion Detection Model', Symmetry, vol. 12, no. 5, Art. no. 5, May 2020, DOI: https://doi.org/10.3390/sym12050754
19. F. Pascale, E. A. Adinolfi, S. Coppola, and E. Santonicola, 'Cybersecurity in Automotive: An Intrusion Detection System in Connected Vehicles', Electronics, vol. 10, no. 15, Art. no. 15, Jan. 2021, DOI: https://doi.org/10.3390/electronics10151765
20. T. Kim and W. Pak, 'Robust network intrusion detection system based on machine-learning with early classification', IEEE Access, vol. 10, pp. 10754–10767, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3145002
21. M. N. Alatawi et al., '[Retracted] Cyber Security against Intrusion Detection Using Ensemble-Based Approaches', Secur. Commun. Netw., vol. 2023, no. 1, p. 8048311, 2023, DOI: https://doi.org/10.1155/2023/8048311
22. İ. Avcı and M. Koca, 'Cybersecurity Attack Detection Model, Using Machine Learning Techniques', Acta Polytech. Hung., vol. 20, no. 7, pp. 29–44, 2023. DOI: https://doi.org/10.12700/APH.20.7.2023.7.2
23. A. Khan and C. Cotton, 'Efficient Attack Detection in IoT Devices using Feature Engineering-Less Machine Learning', ArXiv Prepr. ArXiv230103532, 2023. DOI: https://doi.org/10.48550/arXiv.2301.03532
24. M. A. Alsoufi et al., 'Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review', Appl. Sci., vol. 11, no. 18, Art. no. 18, 2021. DOI: https://doi.org/10.3390/app11188383
25. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, 'Deep learning approach for intelligent intrusion detection system', Ieee Access, vol. 7, pp. 41525–41550, 2019. DOI: https://doi.org/10.1109/ACCESS.2019.2895334
26. A. Dey, 'Deep IDS: A deep learning approach for Intrusion detection based on IDS 2018', in 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), IEEE, 2020, pp. 1–5. Accessed: Sep. 30, 2024. [Online]. Available: DOI: https://doi.org/10.1109/STI50764.2020.9350411
27. Z. Wang, Y. Liu, D. He, and S. Chan, 'Intrusion detection methods based on integrated deep learning model', Comput. Secur., vol. 103, p. 102177, 2021. DOI: https://doi.org/10.1016/j.cose.2021.102177
28. Y. Otoum, D. Liu, and A. Nayak, 'DL-IDS: a deep learning–based intrusion detection framework for securing IoT', Trans. Emerg. Telecommun. Technol., vol. 33, no. 3, p. e3803, Mar. 2022, DOI: https://doi.org/10.1002/ett.3803
29. V. Hnamte and J. Hussain, 'DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system', Telemat. Inform. Rep., vol. 10, p. 100053, 2023. DOI: https://doi.org/10.1016/j.teler.2023.100053
30. A. Henry et al., 'Composition of hybrid deep learning model and feature optimization for intrusion detection system', Sensors, vol. 23, no. 2, p. 890, 2023. DOI: https://doi.org/10.3390/s23020890
31. E. U. H. Qazi, M. H. Faheem, and T. Zia, 'HDLNIDS: hybrid deep-learning-based network intrusion detection system', Appl. Sci., vol. 13, no. 8, p. 4921, 2023. DOI: https://doi.org/10.3390/app13084921
32. V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, 'A novel two-stage deep learning model for network intrusion detection: LSTM-AE', Ieee Access, vol. 11, pp. 37131–37148, 2023. DOI: https://doi.org/10.1109/ACCESS.2023.3266979
33. H. Gonaygunta, G. S. Nadella, P. Pramod Pawar, and D. Kumar, 'Enhancing Cybersecurity: The Development of a Flexible Deep Learning Model for Enhanced Anomaly Detection', in 2024 Systems and Information Engineering Design Symposium (SIEDS), May 2024, pp. 79–84. DOI: https://doi.org/10.1109/SIEDS61124.2024.10534661
34. F. Al-Quayed, Z. Ahmad, and M. Humayun, 'A Situation Based Predictive Approach for Cybersecurity Intrusion Detection and Prevention Using Machine Learning and Deep Learning Algorithms in Wireless Sensor Networks of Industry 4.0', IEEE Access, vol. 12, pp. 34800–34819, 2024, DOI: https://doi.org/10.1109/ACCESS.2024.3372187
35. R. Devendiran and A. V. Turukmane, 'Dugat-LSTM: Deep learning-based network intrusion detection system using chaotic optimization strategy', Expert Syst. Appl., vol. 245, p. 123027, Jul. 2024, DOI: https://doi.org/10.1016/j.eswa.2023.123027
36. Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, 'Benchmarking of machine learning for anomaly-based intrusion detection systems in the CICIDS2017 dataset', IEEE Access, vol. 9, pp. 22351–22370, 2021. DOI:

https://doi.org/10.1109/ACCESS.2021.3056614
37. L. Ashiku and C. Dagli, 'Network Intrusion Detection System using Deep Learning', Procedia Comput. Sci., vol. 185, pp. 239–247, Jan. 2021, DOI: https://doi.org/10.1016/j.procs.2021.05.025
38. M. Asif, S. Abbas, M. A. Khan, A. Fatima, M. A. Khan, and S.-W. Lee, 'MapReduce based intelligent model for intrusion detection using machine learning technique', J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 10, Part B, pp. 9723–9731, Nov. 2022, DOI: https://doi.org/10.1016/j.jksuci.2021.12.008
39. M. A. Kristyanto et al., 'Ssh bruteforce attack classification using machine learning', in 2022 10th International Conference on Information and Communication Technology (ICoICT), IEEE, 2022, pp. 116–119. Accessed: Sep. 30, 2024. [Online]. DOI: https://doi.org/10.1109/ICoICT55009.2022.9914864
40. B. Karthiga, D. Durairaj, N. Nawaz, T. K. Venkatasamy, G. Ramasamy, and A. Hariharasudan, 'Intelligent Intrusion Detection System for VANET Using Machine Learning and Deep Learning Approaches', Wirel. Commun. Mob. Comput., vol. 2022, pp. 1–13, Oct. 2022, DOI: https://doi.org/10.1155/2022/5069104
41. M. U. Ilyas and S. A. Alharbi, 'Machine learning approaches to network intrusion detection for contemporary internet traffic', Computing, vol. 104, no. 5, pp. 1061–1076, May 2022, DOI: https://doi.org/10.1007/s00607-021-01050-5
42. O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, 'EIDM: deep learning model for IoT intrusion detection systems', J. Supercomput., vol. 79, no. 12, pp. 13241–13261, Aug. 2023, DOI: https://doi.org/10.1007/s11227-023-05197-0
43. R. Lazzarini, H. Tianfield, and V. Charissis, 'A stacking ensemble of deep learning models for IoT intrusion detection', Knowl.-Based Syst., vol. 279, p. 110941, 2023. DOI: https://doi.org/10.1016/j.knosys.2023.110941
44. B. Sharma, L. Sharma, C. Lal, and S. Roy, 'Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach', Expert Syst. Appl., vol. 238, p. 121751, Mar. 2024, DOI: https://doi.org/10.1016/j.eswa.2023.121751
45. F. Zhao, H. Li, K. Niu, J. Shi, and R. Song, 'Application of deep learning-based intrusion detection system (IDS) in network anomaly traffic detection', 2024. Accessed: Sep. 30, 2024. [Online]. Available: DOI: https://doi.org/10.20944/preprints202407.0595.v1
46. B. Hade Variant Wahono, Asfihani, I. Mahfud, B. Y. I. Exshadi, and A. M. Shiddiqi, 'Brute Force Detection System Based on Machine Learning Classifier Algorithm in Cloud-Based Infrastructure', in 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Jan. 2024, pp. 939–943. DOI: https://doi.org/10.1109/ICETSIS61505.2024.10459370
47. D. Jim Solomon Raja, R. Sriranjani, P. Arulmozhi, and N. Hemavathi, 'Unified Random Forest and Hybrid Bat Optimization Based Man-in-the-Middle Attack Detection in Advanced Metering Infrastructure', IEEE Trans. Instrum. Meas., vol. 73, pp. 1–12, 2024, DOI: https://doi.org/10.1109/TIM.2024.3420375
48. A. Raj et al., 'Brute forcing on secured shell servers emphasising the role of cyber forensics – a quali-quantitative study', Med. Leg. J., p. 00258172241236269, Jun. 2024, DOI: https://doi.org/10.1177/00258172241236269
49. F. Rustam, A. Raza, M. Qasim, S. K. Posa, and A. D. Jurcut, 'A Novel Approach for Real-Time Server-Based Attack Detection Using Meta-Learning', IEEE Access, vol. 12, pp. 39614–39627, 2024, DOI: https://doi.org/10.1109/ACCESS.2024.3375878
50. Z. e Huma, J. Ahmad, H. A. Hamadi, B. Ghaleb, W. J. Buchanan, and S. U. Jan, 'ACNN-IDS: An Attention-Based CNN for Cyberattack Detection in IoT', in 2024 2nd International Conference on Cyber Resilience (ICCR), Feb. 2024, pp. 1–6. DOI: https://doi.org/10.1109/ICCR61006.2024.10532958
51. D. Shou et al., 'An Intrusion Detection Method Based on Attention Mechanism to Improve CNN-BiLSTM Model', Comput. J., vol. 67, no. 5, pp. 1851–1865, May 2024, DOI: https://doi.org/10.1093/comjnl/bxad105
52. S. Alzughaibi and S. El Khediri, 'A Cloud Intrusion Detection Systems Based on DNN Using Backpropagation and PSO on the CSE-CIC-IDS2018 Dataset', Appl. Sci., vol. 13, no. 4, p. 2276, Jan. 2023, DOI: https://doi.org/10.3390/app13042276
53. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, 'Toward generating a new intrusion detection dataset and intrusion traffic characterization.', ICISSp, vol. 1, pp. 108–116, 2018. DOI: https://doi.org/10.5220/0006639801080116
54. E. Bisong, 'Google Colaboratory', in Building Machine Learning and Deep Learning Models on Google Cloud Platform, Berkeley, CA: Apress, 2019, pp. 59–64. DOI: https://doi.org/10.1007/978-1-4842-4470-8_7
55. M. A. Shyaa, Z. Zainol, R. Abdullah, M. Anbar, L. Alzubaidi, and J. Santamaría, 'Enhanced intrusion detection with data stream classification and concept drift guided by the incremental learning genetic programming combiner', Sensors, vol. 23, no. 7, p. 3736, 2023. DOI: https://doi.org/10.3390/s23073736
56. P. Dini et al., 'Design and testing novel one-class classifier based on polynomial interpolation with application to networking security', IEEE Access, vol. 10, pp. 67910–67924, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3186026
57. Yoo, B. Min, S. Kim, D. Shin, and D. Shin, 'Study on network intrusion detection method using discrete pre-processing method and convolution neural network', IEEE Access, vol. 9, pp. 142348–142361, 2021. DOI: https://doi.org/10.1109/ACCESS.2021.3120839
58. S. Seth, K. K. Chahal, and G. Singh, 'A novel ensemble framework for an intelligent intrusion detection system', IEEE Access, vol. 9, pp. 138451–138467, 2021. DOI: https://doi.org/10.1109/ACCESS.2021.3116219
59. A. Jumabek, S. Yang, and Y. Noh, 'CatBoost-based network intrusion detection on imbalanced CIC-IDS-2018 dataset', 한국통신학회논문지, vol. 46, no. 12, pp. 2191–2197, 2021. DOI: https://doi.org/10.7840/kics.2021.46.12.2191
60. H. Najafi Mohsenabad and M. A. Tut, 'Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS 2018 dataset', Appl. Sci., vol. 14, no. 3, p. 1044, 2024. DOI: https://doi.org/10.3390/app14031044

AUTHORS PROFILE

Nouf Awadh, a dynamic and results-driven IT Services Specialist, has over three years of experience in delivering innovative and effective IT solutions. I hold a bachelor's degree in computer engineering from the College of Computers and Information Technology, Taif University, Taif, SA, and am currently pursuing a master's degree in Cybersecurity. My professional expertise includes identifying and exploiting vulnerabilities, implementing robust security measures, and ensuring IT systems' integrity and performance. Skilled in both independent and collaborative work environments, I thrive in diverse and challenging settings. Committed to continuous learning, I stay updated with advancements in cybersecurity.

Hawazen Zaid, is a dynamic and results-driven IT Services trainee. I hold a bachelor's degree in computer science from the College of Computers and Information Technology, Taif University, Taif, SA, and am currently pursuing a master's degree in Cybersecurity. My professional expertise includes identifying and exploiting vulnerabilities. I thrive in diverse and challenging settings. Committed to continuous learning, I stay updated with advancements in cybersecurity.

Dr. Samah Al-ajmani, received the B.Sc. degree in 2004 and Ph.D. degree in 2019 in King Abdulaziz University, Jeddah, Saudi Arabia, both in Computer Science. She earned the M.Sc. degree in Information Technology from the Queensland University of Technology, Brisbane, Australia. She is currently an Assistance Professor at Taif University, Taif, Saudi Arabia. Her research interests include Cyber Security, AI, IoT, Deep Learning and Machine learning.