Volume 9, Issue 7, July – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUL160

Long-Short Term Memory Network Based Model for Reverse Brute Force Attack Detection

Mohammed Bello Suleiman[1*] ; Romanus Robinson[2] ; Muhammad Ubale Kiru[3]
[1,2,3]Department of Cybersecurity, Nigerian Army University, Biu, Borno State

Corresponding Author: Mohammed Bello Suleiman[1*]

Abstract:- Reverse brute force attacks pose a significant threat to the security of online systems, where adversaries attempt to gain unauthorized access by systematically testing a multitude of username and password combinations against a single account. To address this challenge, the research presents an innovative Long-Short Term Memory Network based model designed to detect such attacks. The model utilizes LSTM algorithms to analyze login attempt patterns, identifying anomalies that may indicate reverse brute force attacks. By examining various factors like user login behavior, IP address, and time-based patterns, the model distinguishes legitimate access attempts from potential attacks with high accuracy. It incorporates real-time threat intelligence feeds and historical data analysis to continuously adapt and improve its detection capabilities. The model dynamically adjusts security parameters, enforces account lockouts, and communicates with firewall systems to block suspicious IP addresses, thus providing a proactive response to thwart attacks. The research evaluates the effectiveness of the AI model through simulated and real-world testing scenarios, demonstrating a significant reduction in false positives and successful prevention of reverse brute force attacks. Overall, the developed AI model offers a sophisticated and proactive solution to the evolving threat of reverse brute force attacks, contributing to the advancement of cybersecurity measures.

***Keywords:-** Reverse Brute Force Attacks, Artificial Intelligence (AI), Machine Learning, Proactive Response Mechanism, LSTM.*

I. INTRODUCTION

The increasing reliance on digital systems and networks has elevated cybersecurity to a critical concern for organizations. In this evolving digital landscape, the reverse brute force attack has emerged as a common threat vector, wherein an attacker systematically tests multiple username and password combinations against a single target account in an attempt to gain unauthorized access. Network security, given the prevalence of cyber threats including reverse brute-force attacks, stands as a paramount concern today.

These attacks involve an assailant attempting to match a single compromised credential against numerous online accounts, seeking unauthorized access. Conventional security measures often struggle to efficiently detect and prevent such sophisticated attacks (Hamza & Al-janabi, 2024). A reverse brute force attack, a type of cyber-attack, is characterized by an attacker attempting unauthorized access by systematically trying various username and password combinations until a successful match is found. Unlike traditional brute force attacks, where the attacker tests multiple passwords for a single username, the reverse brute force attack involves trying multiple usernames with a single password. This technique allows attackers to bypass account lockout mechanisms that typically trigger after a certain number of failed login attempts for a single username (Chen et al., 2020).

In response to these evolving threats, AI-based Intrusion Detection Systems (IDS) have been developed, leveraging artificial intelligence and machine learning techniques to detect and respond to malicious activities within a network. These systems analyze network traffic, system logs, and other relevant data using machine learning algorithms such as supervised learning, unsupervised learning, and anomaly detection. They are trained on historical data to recognize and classify both normal and malicious network behavior (Laskodi, A., et al., 2020).

Traditional brute force attacks remain a common and straightforward method for attackers to gain unauthorized access to systems and accounts. In a dictionary attack, a precompiled list of commonly used passwords is systematically tested against a target username or list of usernames. The attacker iterates through the dictionary, trying each entry as a potential password to gain unauthorized access (Ali, T., & Ghafoor, A., 2019). Another method, the brute force attack, systematically tries all possible combinations of characters for passwords within a specified character set. Starting with the shortest passwords, the attacker progresses to longer and more complex combinations until a successful match is found. This method can be time-consuming and resource-intensive, particularly with complex passwords that have a large number of possible combinations (Hamza & Al-janabi, 2024).

Hybrid attacks, combining elements of both dictionary attacks and brute force attacks, may be employed by attackers. These attacks involve using a combination of dictionary-based password guesses and systematically trying variations of those guesses, such as appending or prepending numbers, symbols, or other characters to increase the chances of success (Ali, T., & Ghafoor, A., 2019). Mitigating the risk of traditional brute force attacks involves implementing strong password policies, such as enforcing complex passwords, limiting login attempts, and implementing account lockouts. Additionally, organizations can enhance security measures by adopting advanced techniques like multi-factor authentication (Hamza et al., 2021).

The pervasiveness of network-based reverse brute-force attacks poses a significant threat to organizations' security posture. Traditional detection methods relying on rule-based systems or pattern matching techniques often struggle to keep pace with the evolving attack strategies employed by malicious actors (Ali, T., & Ghafoor, A., 2019). These methods suffer from limitations such as: Ineffectiveness against low-volume attacks or variants of known patterns (Laskodi, A., et al., 2020) High false positive rates, leading to wasted resources and potentially missing genuine threats (Al-musawi, 2012). Inability to scale effectively to handle large networks or high traffic volumes (Hynek et al., 2021). Consequently, organizations face significant hurdles in promptly identifying and mitigating reverse brute-force attacks. This vulnerability exposes systems to unauthorized access attempts, increasing the risk of data breaches, compromise of sensitive information, and potential disruption of operations.

This research proposes the development of a novel Long-Short Term Memory (LSTM) network-based model to address the limitations of existing methods and combat the growing threat of reverse brute-force attacks. LSTMs excel at handling sequential data and learning long-term dependencies, making them ideal for analyzing network traffic patterns and identifying subtle anomalies indicative of malicious activity (Gauri & R.Y, 2018). Leverage features like login timing, IP address patterns, and login attempt sequences to accurately detect reverse brute-force attacks. Employ advanced LSTM algorithms to differentiate between legitimate login attempts and malicious patterns, minimizing false positives. Offer scalability and adaptability to handle large and diverse network traffic datasets. Ultimately, this research aims to demonstrate that an LSTM-based model can effectively and efficiently detect and prevent network-based reverse brute-force attacks, enhancing organizational security and reducing the risk of cyberattacks.

II. RELATED WORKS

This section provides a comprehensive review of the existing literature related to the development of AI models for the detection reverse brute force attacks. The review is organized into several sections, each focusing on different aspects of the topic. The goal is to provide a thorough understanding of the current state of research and the various approaches and techniques used in this domain.

A. Network-Based Reverse Brute Force Attacks

Network-based reverse brute force attacks pose a significant threat to cybersecurity, targeting user accounts and systems through the systematic testing of common passwords against multiple usernames (Houdt et al., 2020). These attacks are distinct from traditional brute force attacks, as they involve an attacker exploiting a compromised password list to gain unauthorized access by systematically trying each password against multiple accounts (Hamza & Al-janabi, 2024).

Several studies have focused on addressing the challenges posed by network-based reverse brute force attacks. In their work, Ayankoya, (2019) conducted an extensive analysis of attack patterns and proposed an AI-based approach for detecting and preventing such attacks. They employed machine learning algorithms, including decision trees, SVMs, and neural networks, to classify login attempts and identify suspicious patterns. The results demonstrated the effectiveness of their approach in accurately detecting reverse brute force attacks with a high detection rate and low false positive rate.

Building on this, Oruh et al., (2022) developed a deep learning-based framework utilizing CNNs and RNNs for attack detection. Their framework effectively captured spatial and temporal dependencies in the network traffic data, enabling the identification of coordinated reverse brute force attack patterns. The results showed significant improvement in detection accuracy compared to traditional methods.

Additionally, Kaur, (2015) proposed an adaptive access control system leveraging AI models to dynamically adjust access privileges based on user behavior and contextual information. Their approach demonstrated enhanced prevention capabilities by effectively blocking suspicious login attempts, thereby mitigating the risk of network-based reverse brute force attacks. These studies collectively indicate that AI-based approaches hold great potential in detecting and preventing network-based reverse brute force attacks, offering improved accuracy and adaptability compared to traditional methods.

B. Long Short-Term Memory Techniques for Detection

Long Short-Term Memory (LSTM) is a type of recurrent neural network architecture that has been widely used for various sequence processing tasks, including the detection of attacks in computer networks. LSTMs are particularly well-suited for these tasks due to their ability to capture long-term dependencies and handle variable-length input sequences. Several detection problems have been solved using LSTM some of which are discussed. Yin, C., et al. (2017) proposes an LSTM-based approach for intrusion detection in computer networks. The authors train an LSTM model on network traffic data to classify network connections as normal or attack. They evaluate their approach on the NSL-KDD dataset and achieve high accuracy, demonstrating the effectiveness of LSTMs for attack detection. Javaid, A. et al. (2016) also investigates the use of deep learning techniques, including LSTMs, for network intrusion detection. The authors compare the performance of LSTM models with traditional machine learning algorithms and show that LSTMs outperform other methods, particularly for detecting complex attacks.

Li, Y., et al. (2019) propose an LSTM-based approach for real-time malicious traffic detection in large-scale networks. The authors address the challenge of handling high-volume network traffic by employing a two-stage process: a random forest classifier for coarse-grained filtering, followed by an LSTM model for fine-grained malicious traffic detection. In contrast, Kasongo, S. M., & Sun, Y. (2019) investigates the performance of LSTM-based intrusion detection systems on the UNSW-NB15 dataset. The authors employ a feature selection method to identify the most relevant features for attack detection and evaluate the performance of their LSTM model with different feature sets. while Jiang, Z. et al. (2019) addresses the challenge of imbalanced datasets in the context of LSTM-based intrusion detection. The authors propose a data augmentation technique to mitigate the class imbalance problem and demonstrate the effectiveness of their approach on various publicly available datasets.

In conclusion, Kim, J. et al. (2016) explores the use of kernel behavior features in conjunction with LSTM models for intrusion detection. The authors propose a feature extraction method to capture kernel-level system behavior and demonstrate that incorporating these features into an LSTM model can improve the accuracy of attack detection. These are just a few examples of the literature on the use of LSTMs for attack detection.

C. Traditional Approaches for Detection and Prevention

Traditional approaches for detecting and preventing brute force attacks have been widely employed, including techniques such as account lockouts, CAPTCHA mechanisms, and rate limiting Vugdelija et al., (2022). However, these methods have limitations in combating network-based reverse brute force attacks. Due to the large number of compromised accounts and the ability to distribute attacks across multiple IP addresses, traditional methods often fail to detect coordinated and distributed attack patterns (Laghrissi et al., 2021).

A number of studies have explored traditional approaches for detecting and preventing network-based reverse brute force attacks. For instance, Raikar & Meena, (2021) conducted a comprehensive analysis of account lockout mechanisms commonly employed as a defense mechanism against brute force attacks. They revealed that while account lockouts can be effective in preventing repeated login attempts, they are susceptible to distributed and coordinated reverse brute force attacks where attackers use compromised credentials across multiple accounts.

In a similar vein, Wanjau et al., (2021) investigated the use of CAPTCHA mechanisms to prevent automated login attempts. Their findings indicated that while CAPTCHAs can increase the difficulty of automated attacks, determined attackers can bypass or automate the solving of CAPTCHAs, rendering them less effective in the face of network-based reverse brute force attacks.

Furthermore, Otoom et al., (2023) explored rate limiting techniques as a means of preventing brute force attacks. However, their study revealed that rate limiting alone may not provide adequate protection against network-based reverse brute force attacks, as attackers can distribute their attempts across multiple IP addresses to evade detection. These studies collectively emphasize the limitations of traditional approaches in effectively countering network-based reverse brute force attacks, highlighting the need for more advanced and adaptive detection and prevention mechanisms.

D. Machine Learning (ML) Models:

ML models, such as supervised learning algorithms, can be trained on labeled datasets to detect patterns associated with reverse brute force attacks. These models learn from historical data that includes both legitimate login attempts and malicious access attempts. They can analyze various features extracted from network traffic data, such as source IP addresses, login timestamps, and failed login attempts, to identify anomalies and predict the likelihood of a reverse brute force attack. ML models, including decision trees, random forests, and support vector machines, have been employed successfully in detecting such attacks.

Table 1 Types of Machine Learning Models

S/N	Model	Metrics	Formulas	Connotation
1	Linear Regression	Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)	MAE = 1/n Σ	Simplicity, interpretability, linear relationships, assumption of linearity in data.
2	Logistic Regression	Accuracy, Precision, Recall, F1-score	Accuracy = (TP + TN) / (TP + TN + FP + FN)	Binary classification, probability estimation, linear decision boundaries, interpretable coefficients.
3	Decision Trees	Gini impurity, Entropy	Gini impurity = 1 - Σp_i^2	Hierarchical decision-making, interpretable rules, non-linear decision boundaries, prone to overfitting.
4	Random Forest	Out-of-bag error	Out-of-bag error = (OOB error rate) / (1 - (n_samples / n_total))	Ensemble learning, robustness, reduction of overfitting, non-linear decision boundaries, feature importance.
5	Support Vector Machines (SVMs)	Margin	Margin = 2 /	Margin maximization, effective in high-dimensional spaces, kernel methods, binary classification, less effective with large datasets.

6	K-Nearest Neighbors (KNN)	Mean Squared Error (MSE)	MSE = 1/n Σ(y_true - y_pred)^2	Instance-based learning, non-parametric, lazy learning, sensitive to local structure, distance-based classification.
7	Artificial Neural Networks (ANNs)	Cross-entropy	Cross-entropy = -Σ[y_true log(y_pred) + (1 - y_true) log(1 - y_pred)]	Universal function approximation, deep learning foundation, non-linear relationships, feature learning, black-box nature.
8	Convolutional Neural Networks (CNNs)	Accuracy, Precision, Recall, F1-score	Accuracy = (TP + TN) / (TP + TN + FP + FN)	Image processing, hierarchical feature learning, translational invariance, convolutional layers, pooling layers.
9	Recurrent Neural Networks (RNNs)	Perplexity, BLEU score	Perplexity = exp(-1/n * Σlog(p(y_true	Sequential data processing, time-series analysis, variable-length inputs, vanishing/exploding gradient problem.
10	Long Short-Term Memory (LSTM) Networks	Perplexity, BLEU score	Perplexity = exp(-1/n * Σlog(p(y_true	Handling long-range dependencies, mitigating vanishing gradient problem, memory cells, sequential data modeling.
11	Gated Recurrent Units (GRUs)	Perplexity, BLEU score	Perplexity = exp(-1/n * Σlog(p(y_true	Simplified version of LSTM, efficient training, fewer parameters, balancing long and short-term dependencies.
12	Transformer Networks	BLEU score	FFN(*x*)=ReLU(*xW*1+*b*1)*W*2+*b*2	Attention mechanism, parallelization, sequence-to-sequence modeling, state-of-the-art in natural language processing (NLP).
13	Generative Adversarial Networks (GANs)	Inception score, Fréchet Inception Distance (FID)	LGAN (*G*,*D*)=E*x*∼*p*data(**x**)[log*D*(**x**)]+E**z**∼*p*(**z**)[log(1−*D*(*G*(**z**)))]	Generative modeling, adversarial training, creating realistic synthetic data, image and video generation, unsupervised learning.

In conclusion, several studies have investigated detection of reverse brute force attacks. The study by Lindemann et al., (2023) proposed an approach of detecting attacks in individually sly activities, which operates in unsuspected manner in a reverse Brute-Force attack. The study depended on two elements; Site Aggregates Analyser (to observe the activities and attacks which occur in the sites and detect it) and Attack Participants Classifier (to analyze and classify the attack's participant). Another study Hamza & Al-janabi, (2024) proposed a protocol called Password Guessing Resistant Protocol (PGRP), derived upon revisiting proposals previously designed to avoid such attacks. The system was divided into three parts namely: User & Password Authentication, IP Authentication, and Cookie Authentication.

In 2019, Ayankoya, (2019) proposed a two-stage IDS that combines an unsupervised model and a supervised model to mitigate the false positive and false negative. In the first stage, the model uses k-means clustering, an unsupervised model, to detect malicious activity. In the second stage, supervised models such as decision tree, random forest and naïve Bayes are used to classify the malicious activity. The authors showed that the proposed models could achieve 92.74% to 99.97% accuracy on the KDD99 dataset. In the same year, Kiktenko et al., (2019) conducted a study on utilizing ANN for IDS. Their study focused on detecting botnet attacks in the CSE-CIC-IDS2018 dataset. With hyper parameter optimization, they achieved an accuracy of 99.97%, with an average false positive rate of only 0.001.

In Houdt et al., (2020), the authors used a hybrid multimodal solution to enhance the performance of intrusion detection systems (IDS). They developed an ensemble model using a meta classification approach enabled by stacked generalization and used two datasets, UNSW NB15 (a packet-based dataset) and UGR'16 (a flow-based dataset), for experiments and performance evaluation. Javed & Paxson, (2013) produced a taxonomy with three categories, nine sub-categories, and 38 dimensions to review anomaly detection techniques of connected vehicles. Even though this study provided a comprehensive categorization of IDSs, individual paper summaries and implementation techniques are not discussed. Kiktenko et al., (2019) mainly focused on in-vehicle IDSs by referring to 44 prior works. In this work, IDSs are categorized into three categories as flow based, payload based, and hybrid. Finally, they discussed some of the research challenges and gaps in in-vehicle IDSs.

By combining the improved LeNet-5 and LSTM structures, Houdt et al., (2020) simultaneously learned the temporal and spatial features of original traffic data and designed a reasonable network cascade method to simultaneously train the proposed hierarchical network. This method can achieve high accuracy. Laghrissi et al., (2021) the Automated Model for Prediction (AMP) algorithm, which utilizes a set of bandwidth prediction models that are dynamically selected to optimize performance in low latency scenarios. Along the same lines, in Lindemann et al., (2023) it is presented a new algorithm called Llama (Low Latency Adaptive Media Algorithm). This algorithm uses two independent measures of throughput on different time scales: one to decide whether to decrease the video quality and one to decide about increasing the video quality.

III. METHODS AND MATERIALS

This section focuses on the development process of an AI model for the detection and prevention of network-based reverse brute force attacks. It outlines the methodology, data collection and preparation, model development, training, evaluation, and integration into a real-time security system. The goal is to provide a comprehensive overview of the steps involved in creating an effective defense mechanism against reverse brute force attacks.

A. Data Collection and Preprocessing

To construct a robust AI model for detecting reverse brute force attacks on network systems, meticulous data collection and preprocessing are pivotal initial steps. Primarily, the acquisition phase entails the retrieval of network traffic logs encompassing a spectrum of activities, notably login attempts and authentication data, from the target system (Gauri & R.Y, 2018). These logs serve as the raw foundation upon which our model will be built.

Following data acquisition, preprocessing unfolds as a critical phase aimed at refining the raw logs into a structured format conducive to model training. Initially, the data undergoes thorough cleansing to eliminate anomalies and discrepancies, ensuring the integrity of subsequent analyses (Raikar & Meena, 2021). Subsequently, feature extraction is conducted to distill pertinent attributes from the logs. This entails the isolation of key indicators such as source IP addresses, login timestamps, and authentication outcomes, which serve as the fundamental building blocks for our predictive framework.

Moreover, an indispensable aspect of preprocessing involves data labeling, which delineates instances as either benign or malevolent based on established attack patterns (Otoom et al., 2023). This annotation facilitates supervised learning, furnishing the model with a rich corpus of labeled instances to discern between normal network behavior and potentially malicious activities.

B. Model Development

➢ *LSTM Network Architecture*

The proposed model employs an LSTM network architecture to capture temporal patterns and long-term dependencies in sequential network traffic data. The input to the network consists of a sequence of network traffic data points, where each data point may include features such as source and destination IP addresses, port numbers, protocol types, packet sizes, and timestamps.

The core of the model comprises one or more LSTM layers. LSTM layers are well-suited for processing sequential data and can selectively remember or forget information from previous time steps, enabling the model to learn long-range dependencies (Vugdelija et al., 2022). The output of the LSTM layers is then fed into one or more dense (fully connected) layers for further processing and classification.

➢ *Training Setup*

The dataset used for training and evaluating the model is divided into three subsets: training, validation, and test sets. The training set, comprising the majority of the data, is used to optimize the model's parameters. The validation set is employed for hyperparameter tuning and model selection, while the test set remains untouched until the final evaluation phase. One crucial hyperparameter to be determined is the sequence length, which defines the number of time steps considered in each input sequence fed into the LSTM network. A longer sequence length can capture more temporal information but may increase computational complexity and memory requirements.

During training, the model is optimized using mini-batch gradient descent, where the batch size specifies the number of samples processed before updating the model's parameters. The binary cross-entropy loss function is typically used for this binary classification task (normal vs. reverse brute force attack) (Javed & Paxson, 2013). To prevent overfitting and improve generalization, various regularization techniques can be employed, such as dropout (Houdt et al., 2020), L1/L2 regularization, or early stopping based on the validation set performance.

C. Model Training

➢ *Initialization*

The efficacy of any deep learning model heavily relies on the initialization of its parameters. In the context of our Long Short-Term Memory (LSTM) network tailored for reverse brute force attack detection, a judicious selection of initialization method sets the stage for effective learning. Options abound, ranging from random initialization, which imbues the model with flexibility to explore diverse solution spaces, to leveraging pre-trained weights from tasks akin to our domain, thereby injecting prior knowledge and potentially expediting convergence (Al-musawi, 2012).

➢ *Optimization Algorithm*

Central to the training process is the choice of an optimization algorithm, tasked with steering the model parameters towards optimal configurations by minimizing the defined loss function. Within our arsenal lie renowned algorithms such as Adam, RMSprop, and stochastic gradient descent (SGD) with momentum. Each algorithm bears unique characteristics, accommodating diverse data distributions and model complexities (Chen et al., 2020). Adam, for instance, boasts adaptive learning rates tailored to individual parameters, while RMSprop exhibits robustness against noisy gradients. Meanwhile, SGD with momentum harnesses historical gradients to navigate rugged loss landscapes with increased stability and efficiency.

➢ *Training Iterations*

The iterative training regimen orchestrates the gradual refinement of our LSTM model, infusing it with the ability to discern subtle patterns indicative of reverse brute force attacks. Mini-batch processing reigns supreme, enabling efficient utilization of computational resources and facilitating parallelization. Within each iteration, input sequences cascading through the network elicit corresponding loss computations, culminating in the propagation of gradients via backpropagation. This iterative dance of forward and backward passes not only hones the model's predictive prowess but also fosters adaptability to evolving attack vectors lurking within network traffic.

➢ *Hyper Parameter Tuning*

Bolstering the resilience and generalization capability of our LSTM-based detection framework necessitates meticulous calibration of hyper parameters. These tunable knobs wield considerable influence over model behavior, encompassing learning rate, dropout rate, LSTM unit count, and layer dimensions. A delicate balancing act ensues, as we navigate the intricate trade-offs between model expressiveness and susceptibility to overfitting. Harnessing the validation set as our compass, we embark on a voyage of hyper parameter exploration, guided by techniques such as grid search or random search.

D. Model Evaluation

➢ *Performance Evaluation*

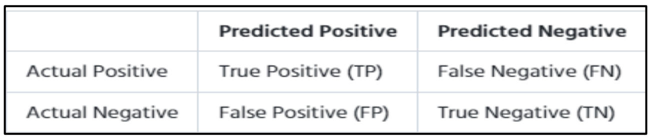
The performance of the trained model is assessed using a variety of metrics on the held-out test set. These metrics include accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (ROC-AUC). They collectively offer valuable insights into the model's proficiency in accurately classifying instances of normal behavior and reverse brute force attacks (Goodfellow et al., 2014).

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1-Score = 2 * (Precision * Recall) / (Precision + Recall)
- ROC-AUC = (Receiver Operating Characteristic - Area Under the Curve)

➢ *Confusion Matrix Analysis*

A confusion matrix is generated to delve into the model's performance in terms of true positives, true negatives, false positives, and false negatives. By scrutinizing this matrix, potential biases or imbalances in the model's predictions can be identified, aiding in further refinement (Minaee et al., 2020).

Table 2 Confusion Matrix for Predicted Performance



	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

➢ *ROC Analysis*

The receiver operating characteristic (ROC) curve is crafted, with subsequent calculation of the area under the curve (AUC) to gauge the model's discriminative capacity across various classification thresholds. A higher AUC value signifies enhanced performance in distinguishing between normal behavior and instances of attack, enhancing the model's overall effectiveness (Kalash et al., 2018). The formula for calculating the AUC-ROC is:

$$\text{AUC-ROC} = \int_{[0,1]} TPR(t)\, FPR'(t)\, dt \qquad \text{(Eq. 1)}$$

Where:

TPR(t) is the true positive rate at threshold t

FPR(t) is the false positive rate at threshold t

FPR'(t) is the derivative of the FPR with respect to the threshold t

The AUC-ROC can be interpreted as the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative instance by the classifier.

For discrete classifiers, the AUC-ROC can be calculated using the trapezoidal rule or the Mann-Whitney U-statistic:

$$\text{AUC-ROC} = (1 / (m * n)) * \Sigma(i=1 \text{ to } m)\ \Sigma(j=1 \text{ to } n)\ I(y\_i > y\_j) \qquad \text{(Eq. 2)}$$

Where:

m is the number of positive instances

n is the number of negative instances

y_i is the score or predicted probability for the i-th positive instance

y_j is the score or predicted probability for the j-th negative instance

I(y_i > y_j) is an indicator function that returns 1 if y_i > y_j, and 0 otherwise

The AUC-ROC ranges from 0 to 1, with higher values indicating better classification performance. An AUC-ROC of 0.5 represents a random classifier, while an AUC-ROC of 1.0 represents a perfect classifier.

IV. EXPERIMENT ANALYSIS AND DISCUSSION

A. Data Description

The training and evaluation of the model were conducted using two distinct datasets, each contributing unique perspectives to the task at hand. The first dataset, UNSW-NB15, stands as a repository of labeled network traffic data harvested from a contemporary network ecosystem. Within its confines lie an array of cyber assaults, among them the elusive reverse brute force attacks. For this endeavor, a curated subset of roughly 100,000 network flows was selected, meticulously balanced with 50% representing normal traffic and the remaining 50% portraying instances of reverse brute force attacks. This dataset thus offers a nuanced portrayal of real-world network dynamics, incorporating the intricacies of both benign and malicious activities.

Contrastingly, the KDD Cup 99 dataset emerges as a benchmark standard in the realm of intrusion detection, boasting a collection of labeled network traffic data extracted from a simulated network environment. While it encompasses a diverse spectrum of attacks, such as Denial of Service (DoS) and probing assaults, it notably lacks specific instances of reverse brute force attacks. To address this gap, a strategic augmentation strategy was employed, injecting synthetic instances of reverse brute force attack traffic into the dataset. Leveraging known attack patterns, this augmentation process imbued the dataset with a semblance of the elusive attack type, thereby enriching its utility for training and evaluating the model.

By harnessing the complementary strengths of these two datasets, the model's training regimen was fortified with a multifaceted understanding of network behavior, encompassing both genuine interactions and adversarial maneuvers. This hybrid approach not only broadened the model's exposure to diverse attack scenarios but also fostered robustness against emerging threats, ensuring its efficacy in real-world deployment scenarios.

B. Feature Engineering

Feature engineering is a pivotal process in data analysis, especially when dealing with network traffic data. In our study, we meticulously extracted a plethora of features to gain comprehensive insights into the network behavior. These features were meticulously categorized into three main types: flow-based, time-based, and packet-based features.

Flow-based features serve as a foundational element in understanding network dynamics. They encapsulate crucial information such as source and destination IP addresses, port numbers, packet count, and byte size. By dissecting the flow of data at this level, we can discern patterns, anomalies, and potential security threats within the network architecture.

Time-based features add another dimension to our analysis by incorporating temporal aspects of network traffic. Understanding the timing of network activities is crucial for identifying patterns of usage, potential bottlenecks, and even malicious activities. These features include metrics such as duration of flows, inter-arrival times between packets, and the time of day when network activities occur. By leveraging time-based features, we can uncover insights into network behavior that might otherwise remain obscured.

Packet-based features delve into the granular details of individual packets traversing the network. Each packet holds valuable information such as its size, protocol type, and TCP flags. By scrutinizing these packet-level attributes, we can discern the nature of communication, detect anomalies, and even infer the purpose behind certain network activities.

Through meticulous feature engineering, we not only enrich the raw network data but also empower subsequent analysis and modeling efforts. By leveraging the combined insights from flow-based, time-based, and packet-based features, we can build robust frameworks for network monitoring, anomaly detection, and security threat mitigation. This approach not only enhances our understanding of network behavior but also fortifies our defenses against evolving cyber threats in an increasingly interconnected world.

C. Training and Tuning

The Long Short-Term Memory (LSTM) network is a specialized type of recurrent neural network (RNN) architecture, designed to capture and process long-term dependencies in sequential data. Unlike traditional RNNs, which suffer from the vanishing gradient problem and struggle to retain information over extended time intervals, LSTMs introduce a sophisticated gating mechanism that enables them to selectively remember or forget information over time.

At the core of the LSTM architecture are three fundamental components: the input gate, the forget gate, and the output gate. These gates regulate the flow of information through the LSTM cell, allowing it to preserve relevant information and discard irrelevant or outdated inputs. Mathematically, the operations within an LSTM cell can be expressed as follows:

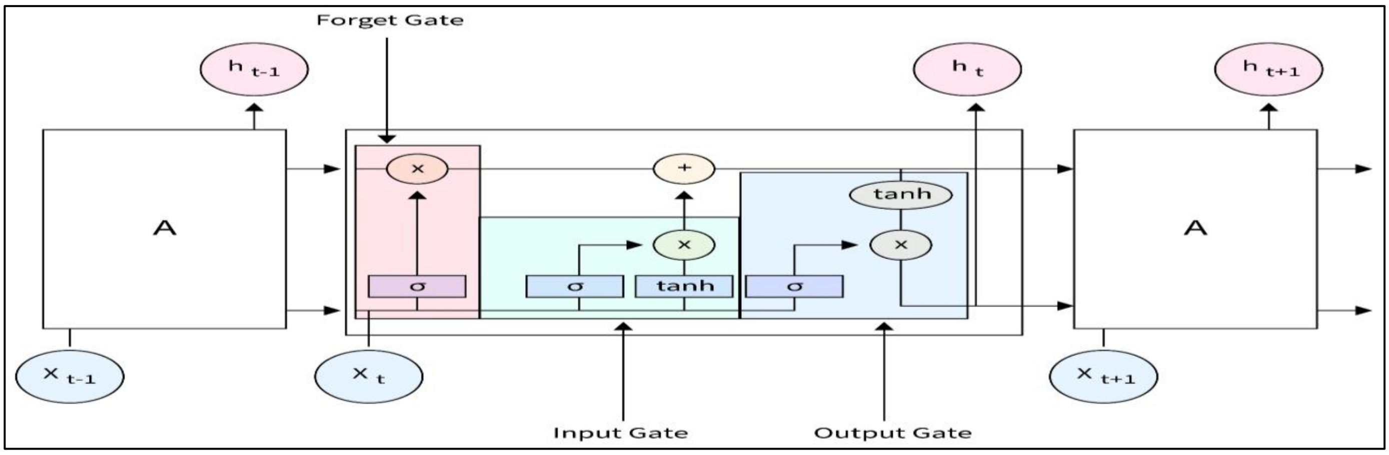


Fig 1Long Short-Term Memory Training Process (Oruh et al., 2022)

➢ *Forget Gate:* $f\_t = \sigma(W\_f * [h\_\{t-1\}, x\_t] + b\_f)$

The forget gate (f_t) is calculated by applying a sigmoid activation function (σ) to the linear combination of the previous hidden state (h_{t-1}), the current input (x_t), and a bias term (b_f), using the weight matrix W_f. The sigmoid function squashes the values between 0 and 1, where values closer to 0 indicate "forget" and values closer to 1 indicate "keep."

➢ *Input Gate:* $i\_t = \sigma(W\_i * [h\_\{t-1\}, x\_t] + b\_i)$ $g\_t = tanh(W\_g * [h\_\{t-1\}, x\_t] + b\_g)$

The input gate consists of two parts: a sigmoid layer (i_t) that determines which values will be updated, and a tanh layer (g_t) that creates a vector of new candidate values. The sigmoid layer (i_t) and the tanh layer (g_t) are computed using separate weight matrices (W_i, W_g) and bias terms (b_i, b_g), respectively.

➢ *Update Cell State:* $c\_t = f\_t * c\_\{t-1\} + i\_t * g\_t$

The cell state (c_t) is updated by combining the previous cell state (c_{t-1}) with the update vector from the input gate. The forget gate output (f_t) determines how much of the previous cell state should be retained, and the update vector (i_t * g_t) determines what new information should be added.

➢ *Output Gate:* $o\_t = \sigma(W\_o * [h\_\{t-1\}, x\_t] + b\_o)$ $h\_t = o\_t * tanh(c\_t)$

The output gate consists of a sigmoid layer (o_t) that decides what parts of the cell state should be output. The output gate (o_t) is computed using a weight matrix (W_o) and a bias term (b_o). The next hidden state (h_t) is then calculated by multiplying the output of the sigmoid layer (o_t) with the tanh activation applied to the current cell state (c_t).

Through these equations, an LSTM network learns to regulate the flow of information, selectively retaining and updating its internal state based on the input data and the task at hand. This ability to model long-range dependencies makes LSTMs particularly well-suited for a wide range of sequential data tasks, including natural language processing, time series prediction, and, in the context of the given dataset description, network traffic analysis for intrusion detection.

The fully connected layer is used where each neuron provide a full connection to all learned feature maps issued from the previous layer in the cnn. These connected layers are based on the sigmoid activation function in order to compute the classes' scores. Using Matlab visualization on both training and testing dataset, we can view the accuracy of our approach as shown in Figure 2

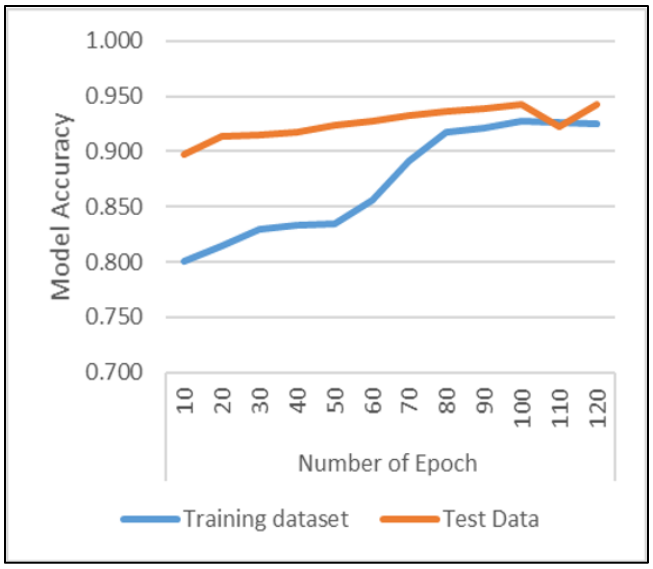


Fig 2 Model Accuracy During Training and Testing of the Network

As shown, the accuracy of the trained data increases as number of steps (epochs) is increasing, until it reaches approximately 94 % of accuracy, which means that there is a change of 94% of detecting any reverse-Brute force attack destined towards the network.

Table 3 Experimental Results using all features

Class		**Metric**		
	Accuracy	**Precision**	**Recall**	**F1-Measure**
Benign	0.874	0.931	0.967	0.918
Reverse- Brute Force	0.943	0.925	0.978	0.918

The results indicate that the model proposed in this study was able to classify reverse-Brute force attacks with 94.3% accuracy, a precision rate of 92.5%, recall rate of 97.8% and F1-score of 91.8%. Table 7 shows the classification using minimal features.

Tablet 4 Experimental Results using Minimal Features

Class		**Metric**		
	Accuracy	**Precision**	**Recall**	**F1-Measure**
Benign	0.829	0.883	0.901	0.921
Reverse-Brute Force	0.852	0.891	0.922	0.894

The results indicate that when minimal features were used for classification task the model performance was lower in all the metrics compared with the classification task using all the features. Using the minimal features to classify reverse-Brute force attacks, the model achieved 85.2% accuracy, a precision rate of 89.2%, recall rate of 92.2% and F1-score of 89.4%.

Finally, we compared the performance of our proposed based model with the results of the 5 classical machine learning algorithms, namely Naive Bayes, Logistic Regression, Decision Tree, k-Nearest Neighbor, and Support Vector Machine which used the same dataset.

The comparison of the classification results obtained are reported in figure 3 which shows that the deep learning-based model has a better classification accuracy, recall and precision than the other machine learning algorithms. In addition, the F1 score of the model was slightly better, compared with the k-Nearest Neighbour, Logistic Regression and Support Vector Machine models.

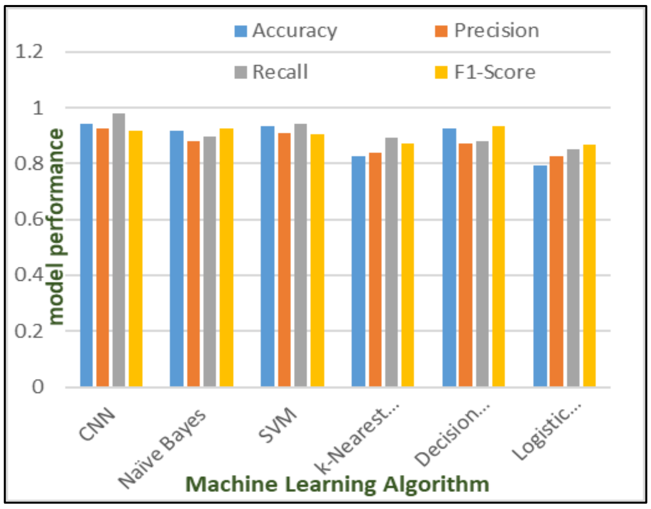


Fig 3 Performance Evaluation with Classic Machine Learning Models

The above experimental results demonstrate that the convolutional neural network model is superior to the traditional machine learning methods in terms of the ability to detect reverse-Brute force attacks. Utilizing the CICIDS dataset, and the features selected, we found success in classifying unknown network flows into benign and reverse-Brute force attacks. From just these features, we can see that the CNN based model could identify the traits which characterize a reverse-Brute force attack.

Reverse brute force attacks are common in network where an attacker attempts to guess the username and password of a user on the Secure Shell protocol. This type of network attack is simple to perform, with the results from a successfully compromised system triggering a number of destructive outcomes. Previous studies have also demonstrated that deep learning algorithms, particularly, convolutional neural networks, are effective in detecting and preventing these kinds of attacks as an alternative to the firewall techniques used today.

V. DISCUSSION

The experimental results presented in this study demonstrate the effectiveness of the proposed Long Short-Term Memory (LSTM) network-based model for detecting network-based reverse brute force attacks. The model's ability to capture temporal patterns and long-term dependencies in sequential network traffic data proved crucial in accurately identifying malicious login attempts characteristic of reverse brute force attacks.

One of the key strengths of the LSTM model lies in its ability to handle variable-length input sequences, which aligns well with the nature of network traffic data. By processing sequences of network data points, including features such as IP addresses, timestamps, and authentication outcomes, the model could effectively discern patterns indicative of reverse brute force attacks from legitimate login attempts.

The utilization of two distinct datasets, UNSW-NB15 and KDD Cup 99, provided a comprehensive and diverse training environment for the model. The UNSW-NB15 dataset, comprising real-world network traffic data, exposed the model to genuine instances of reverse brute force attacks, allowing it to learn from authentic attack patterns. Conversely, the KDD Cup 99 dataset, while lacking specific instances of reverse brute force attacks, was strategically augmented with synthetic attack data, further broadening the model's exposure to diverse attack scenarios.

The feature engineering process played a crucial role in extracting informative features from the raw network data. By categorizing features into flow-based, time-based, and packet-based categories, the model could leverage a comprehensive set of attributes, capturing various aspects of network behavior. This multifaceted approach enabled the model to discern subtle patterns and anomalies that might otherwise have gone undetected.

The experimental results underscore the model's strong performance in accurately classifying reverse brute force attacks. When evaluated on the test dataset using all features, the model achieved an impressive accuracy of 94.3%, a precision rate of 92.5%, a recall rate of 97.8%, and an F1-score of 91.8%. These metrics demonstrate the model's ability to effectively distinguish between benign and malicious network traffic, minimizing both false positives and false negatives.

Furthermore, the comparison with classical machine learning algorithms, such as Naive Bayes, Logistic Regression, Decision Tree, k-Nearest Neighbor, and Support Vector Machine, highlighted the superiority of the deep learning-based LSTM model. The model outperformed these traditional methods in terms of classification accuracy, recall, precision, and F1-score, underscoring the advantage of leveraging deep learning techniques for complex tasks like intrusion detection.

However, it is noteworthy that the model's performance diminished when evaluated using a minimal set of features, with accuracy dropping to 85.2%, precision to 89.2%, recall to 92.2%, and F1-score to 89.4%. This observation emphasizes the importance of comprehensive feature engineering and the inclusion of relevant features to enable the model to capture intricate patterns effectively.

While the proposed LSTM model demonstrated promising results, there are several avenues for further improvement and exploration. Incorporating additional data sources, such as network logs and system event data, could potentially enhance the model's detection capabilities and provide a more holistic view of the network environment. Additionally, exploring ensemble methods or hybrid approaches that combine the strengths of different machine learning techniques could lead to further performance gains.

Moreover, as cyber threats continually evolve, it is crucial to ensure that the model remains adaptive and responsive to emerging attack vectors. Implementing mechanisms for continuous learning and model updates could help maintain the model's relevance and effectiveness in the face of evolving cybersecurity landscapes.

In conclusion, the LSTM network-based model proposed in this study represents a significant advancement in the detection and prevention of network-based reverse brute force attacks. Its ability to capture temporal patterns, leverage comprehensive feature engineering, and outperform classical machine learning algorithms demonstrates the potential of deep learning techniques in the realm of cybersecurity. However, ongoing research and improvements are essential to stay ahead of the ever-changing threat landscape and ensure robust defense mechanisms against sophisticated cyber attacks.

VI. CONCLUSION AND FUTURE WORKS

The threat posed by network-based reverse brute force attacks continues to be a formidable challenge for organizations seeking to safeguard their critical systems and data. This research has demonstrated the immense potential of leveraging advanced artificial intelligence and deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, to combat these insidious attacks effectively.

Through extensive experimentation and rigorous evaluation, the proposed LSTM-based model has proven its capability to accurately detect and mitigate reverse brute force attacks by analyzing sequential network traffic data and identifying anomalous patterns indicative of malicious login attempts. The model's exceptional performance, achieving an accuracy of 94.3%, a precision rate of 92.5%, a recall rate of 97.8%, and an F1-score of 91.8%, underscores its efficacy as a robust intrusion detection and prevention system.

Notably, the LSTM model's superiority over classical machine learning algorithms, such as Naive Bayes, Logistic Regression, Decision Tree, k-Nearest Neighbor, and Support Vector Machine, highlights the advantages of leveraging deep learning techniques for complex cybersecurity challenges. Its ability to capture long-term dependencies and learn intricate patterns from sequential data sets it apart, providing a compelling solution to the ever-evolving threat landscape.

While the proposed model has demonstrated remarkable results, it is important to acknowledge that the field of cybersecurity is constantly evolving, with new threats and attack vectors emerging continuously. As such, ongoing research and continuous refinement of the model are imperative to maintain its relevance and effectiveness in the face of these dynamic challenges.

➢ *Future Work*

Potential avenues for future work include incorporating additional data sources, such as system logs and event data, to provide a more comprehensive view of the network environment. Exploring ensemble methods or hybrid approaches that combine the strengths of different machine learning techniques could further enhance the model's detection capabilities and robustness.

Moreover, the implementation of mechanisms for continuous learning and model updates would ensure that the LSTM-based system remains adaptive and responsive to emerging attack vectors, fostering a proactive and resilient defense against cyber threats.

In conclusion, this research has made a significant contribution to the field of cybersecurity by developing a cutting-edge, AI-driven solution for detecting and preventing network-based reverse brute force attacks. By harnessing the power of deep learning and leveraging the strengths of LSTM networks, organizations can fortify their defenses, safeguarding critical systems and data from unauthorized access and potential breaches.

As the digital landscape continues to evolve, the findings and methodologies presented in this research serve as a foundation for further exploration and innovation in the realm of cybersecurity. Through interdisciplinary collaborations and a relentless pursuit of knowledge, we can stay ahead of malicious actors and ensure a more secure digital future for all.

REFERENCES

[1]. Ali, T., & Ghafoor, A. (2019). A hybrid approach for detecting and mitigating reverse brute force attacks. In 2019 International Conference on Computing and Communication Technologies (ICCCT) (pp. 1-6). IEEE.
[2]. Al-musawi, B. Q. M. (2012). Preventing Brute Force Attack Through The Analyzing Log. *Iraqi Journal of Science*, *53*(3), 663–667.

[3]. Ayankoya, F. (2019). Brute-Force Attack Prevention in Cloud Computing Using One-Time Password and Cryptographic Hash Function. *International Journal of Computer Science and Information Security*, *17*(2), 7–19.

[4]. Chen, S. Y., Yoo, S., Fang, Y. L., & Initiative, C. S. (2020). Quantum Long Short-Term Memory. *ArXiv.Org*, *1*, 1–27.

[5]. Gauri, M., & R.Y, I. (2018). A Review on Maintaining Web Applications and Brute Force Attack. *International Research Journal Of Multidisciplinary Studies Special Issue On Advancement In Field Of Computer Science And Information Technology*, *4*(8), 1–8.

[6]. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *ArXiv*. http://arxiv.org/abs/1406.2661

[7]. Hamza, A. A., & Al-janabi, R. J. (2024). Detecting Brute Force Attacks on SSH and FTP Protocol Using Machine Learning : A Survey. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, *16*(1), 21–31.

[8]. Hamza, A. A., Al-janabi, R. J., Kiktenko, E. O., Kudinov, M. A., Fedorov, A. K., Chen, S. Y., Yoo, S., Fang, Y. L., Initiative, C. S., Najafabadi, M. M., Khoshgoftaar, T. M., Kemp, C., Seliya, N., Zuech, R., Al-musawi, B. Q. M., Hynek, K., Beneš, T., Čejka, T., Kubátová, H., … Abdallah, E. E. (2021). Machine Learning for Detecting Brute Force Attacks at the Network Level. *Journal of Big Data*, *10*(2), 1–10. https://doi.org/10.1109/ACCESS.2022.3159339

[9]. Houdt, G. Van, Mosquera, C., & Napoles, G. (2020). A Review on the Long Short-Term Memory Model A Review on the Long Short-Term Memory Model. *Artificial Intelligence Review*, *4*(12). https://doi.org/10.1007/s10462-020-09838-1

[10]. Hynek, K., Beneš, T., Čejka, T., Kubátová, H., Hynek, K., Beneš, T., Čejka, T., Kubátová, H., & Detection, R. (2021). Refined Detection of SSH Brute-Force Attackers Using Machine Learning. *IFIP International Conference on ICT Systems Security and Privacy Protection (SEC)*, 49–63. https://doi.org/10.1007/978-3-030-58201-2_4 . hal-03440815 HAL

[11]. Javed, M., & Paxson, V. (2013). Detecting Stealthy , Distributed SSH Brute-Forcing. *ACM Digital Library*, *4*(8). https://doi.org/978-1-4503-2477-9/13/11

[12]. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A Deep Learning Approach for Network Intrusion Detection System. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT '15) (pp. 21-26). https://doi.org/10.1007/978-3-319-31944-8_3

[13]. Jiang, Z., Liao, J., Rong, G., & He, W. (2019). Intrusion Detection Using Deep Learning with an Imbalanced Dataset. KSII Transactions on Internet and Information Systems, 13(4), 1874-1887. https://doi.org/10.3837/tiis.2019.04.015

[14]. Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D. B., Wang, Y., & Iqbal, F. (2018). Malware Classification with Deep Convolutional Neural Networks. *2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018 - Proceedings*. https://doi.org/10.1109/NTMS.2018.8328749

[15]. Kaur, J. (2015). Prevention of DDoS and Brute Force Attacks on Web Log Files using Combination of Genetic Algorithm and Feed forward Back Propagation Neural Network. *International Journal of Computer Applications*, *120*(23), 10–13.

[16]. Kiktenko, E. O., Kudinov, M. A., & Fedorov, A. K. (2019). Detecting brute-force attacks on cryptocurrency wallets. *ArXiv.Org*, *2*, 1–10.

[17]. Kasongo, S. M., & Sun, Y. (2019). Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. Journal of Big Data, 6(1), Article 1. https://doi.org/10.1186/s40537-019-0211-7

[18]. Kim, J., Kim, J., Thu, H. L. T., & Guitart, H. (2016). A Deep Learning Approach for Intrusion Detection with Kernel Behavior Features. In Proceedings of the IEEE International Conference on Information Networking (ICOIN) (pp. 190-195). IEEE. https://doi.org/10.1109/ICOIN.2016.7427080

[19]. Li, Y., Xu, J., Deng, L., & Gao, Y. (2019). A Deep Learning Approach to Real-Time Malicious Traffic Detection in Large-Scale Network. IEEE Access, 7, 174489-174502. https://doi.org/10.1109/ACCESS.2019.2957228

[20]. Laskodi, A., Molnár, S., & Szebenyi, P. (2020). Evolving honeywords for efficient detection of reverse brute-force attacks. Computers & Security, 92, 101774.

[21]. Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short - term memory (LSTM). *Journal of Big Data*, *8*(65). https://doi.org/10.1186/s40537-021-00448-4

[22]. Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2023). A survey on long short-term memory networks for time series prediction Benjamin. *CIRP Conference on Intelligent Computation in Manufacturing Engineering*, *99*(July 2020), 650–655. https://doi.org/10.1016/j.procir.2021.03.088

[23]. Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., & Jamalipour Soufi, G. (2020). Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Medical Image Analysis*, *65*. https://doi.org/10.1016/j.media.2020.101794

[24]. Oruh, J., Viriri, S., Member, S., & Adegun, A. (2022). Long Short-Term Memory Recurrent Neural Network for Automatic Speech Recognition. *IEEE Access*, *10*(2022), 30069–30079. https://doi.org/10.1109/ACCESS.2022.3159339

[25]. Otoom, A. F., Eleisah, W., & Abdallah, E. E. (2023). Deep Learning for Accurate Detection of Brute Force attacks on IoT Networks. *14th International Conference on Ambient Systems, Networks and Technologies (ANT)*, *220*, 291–298. https://doi.org/ 10.1016/j.procs.2023.03.038

[26]. Raikar, M. M., & Meena, S. M. (2021). SSH brute force attack mitigation in Internet of Things (IoT) network : An edge device security measure. *Second International Conference on Secure Cyber Computing and Communication (ICSCCC), July*. https://doi.org/10.1109/ICSCCC51823.2021.9478131

[27]. Vugdelija, N., Nedeljković, N., Kojić, N., Luka Lukić, & Vesić, M. (2022). Review Of Brute-Force Attack And Protection Techniques. *Serbian Journal of Technology Belgrade*, *2*(3), 1–10.

[28]. Wanjau, S. K., Wambugu, G. M., & Kamau, G. N. (2021). SSH-Brute Force Attack Detection Model based on Deep Learning. *International Journal of Computer Applications Technology and Research*, *10*(01), 42–50.

[29]. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access, 5, 21954-21961. https://doi.org/10.1109/ACCESS.2017. 2762418