International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-13 Issue-6, March 2025

An Algorithm for Detecting Brute Force Attacks on FTP and SSH Services Utilizing Deep Learning with Probabilistic Neural Networks (PNN)

Hanadi Alosimy, Jawaher Alzaidi, Samah Alajmani, Ben Soh

Abstract: ***Brute force attacks remain one of the most prevalent and effective methods cybercriminals use to gain unauthorized access to networks and systems. These attacks involve systematically attempting various password or key combinations until the correct one is identified, often targeting critical services such as FTP (File Transfer Protocol) and SSH (Secure Shell). The consequences of these attacks can be severe, including data breaches, financial losses, and reputational damage. Intrusion Detection Systems (IDS) play a crucial role in mitigating these threats by monitoring network traffic and identifying malicious activities. However, traditional IDS methods—such as signature-based detection and anomaly detection—struggle to detect emerging and evolving threats. To address these challenges, this study presents an advanced detection model that utilises deep learning techniques, specifically a Probabilistic Neural Network (PNN), to identify brute-force attacks on FTP and SSH protocols. The model is trained and evaluated using the CICIDS2018 dataset, with the Bat Optimization Algorithm employed to fine-tune parameters and enhance performance. The proposed model achieves remarkable results, with an accuracy of 99.968%, precision of 99.949%, recall of 99.986%, and an F1-score of 99.968%. These findings highlight the model's potential as a highly effective tool for strengthening network security and preventing unauthorized access.***

Keywords: BAT Optimization Algorithm, Brute Force, Deep Learning, IDS, PNN.

I. INTRODUCTION

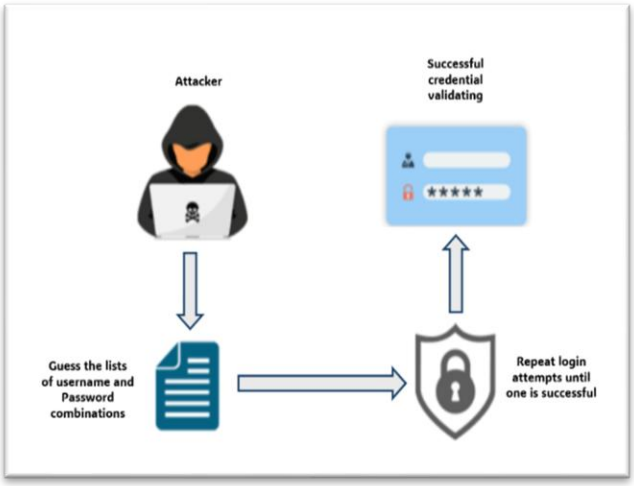
The increasing complexity and volume of network traffic underscore the need for advanced Intrusion Detection Systems (IDS) to identify and mitigate cyber threats effectively.

Manuscript received on 12 November 2024 | Revised Manuscript received on 28 December 2024 | Second Revised Manuscript received on 16 January 2025 | Manuscript Accepted on 15 March 2025 | Manuscript published on 30 March 2025.
*Correspondence Author(s)
Hanadi Alosimy, Department of Information Technology, College of Computer and Information Technology, Taif University, Saudi Arabia. Email ID: hanadiosimy@gmail.com, ORCID ID: 0009-0009-1690-8439
Jawaher AlZaidi*, Department of Information Technology, College of Computer and Information Technology, Taif University, Saudi Arabia. Email ID: jawaher52477@gmail.com, ORCID ID: 0009-0001-8256-1850
Samah H. Alajmani, Department of Information Technology, College of Computer and Information Technology, Taif University, Saudi Arabia. Email ID: s.ajmani@tu.edu.sa, ORCID ID: 0009-0000-7152-9559
Ben Soh, Department of Computer Science and Computer Engineering, La Trobe University, Bundoora, Australia; Email ID: B.soh@latrobe.edu.au. ORCID ID: 0000-0002-9519-886X

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

These attacks involve systematically attempting multiple combinations of encryption keys or passwords until the correct one is identified, as illustrated in Fig. 1 [1].

Such attacks pose severe risks to both organizations and individuals, potentially resulting in data breaches, identity theft, and significant financial losses. As network traffic continues to grow and attackers refine their techniques, traditional IDS approaches struggle to keep pace, highlighting the need for more sophisticated detection methodologies.



[Fig.1: Brute Force Attack Method [1]]

The paper is organized as follows: The second section provides the background about IDS, the third section reviews the literature and previous works, the fourth section discusses research methodology, the fifth section discusses the proposed model, the sixth section discusses requirement analysis, the seventh section discusses experiment and result analysis, and the eighth section discusses the conclusion and future work.

II. BACKGROUND

An Intrusion Detection System (IDS) is designed to identify malicious and unauthorized activities by continuously monitoring network traffic. It also aids network administrators in implementing preventive measures to protect network infrastructure and its connected devices [2]. Traditionally, IDS solutions have relied on two primary detection approaches [3]:

© Copyright: All rights reserved.

A. Signature-Based Detection: This method identifies attacks based on predefined patterns or signatures of known threats. While effective in detecting previously documented attacks, it struggles to recognize novel or evolving threats [3]. Figure 2 illustrates the Signature-Based Detection Flow.

B. Anomaly-Based Detection: This approach identifies deviations from expected network behavior. While it has the potential to detect unknown threats, it is prone to false positives, particularly in dynamic environments where user behavior frequently changes [3]. Figure 3 presents the Anomaly-Based Detection Flow.

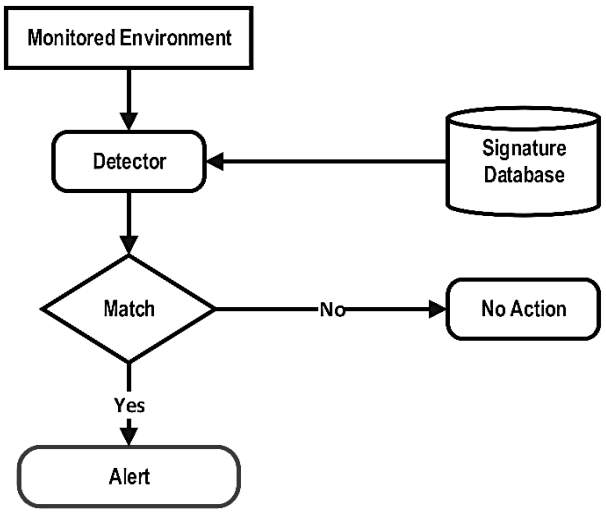
Despite their effectiveness, traditional IDSs face challenges in detecting brute force attacks, including:

- **Evasion Techniques:** Attackers employ sophisticated techniques, such as distributed brute force attacks, which make detection more difficult for conventional IDS solutions.
- **High False Positive Rates:** Excessive false positives can overwhelm security professionals, leading to alert fatigue and an increased likelihood of missing critical threats.

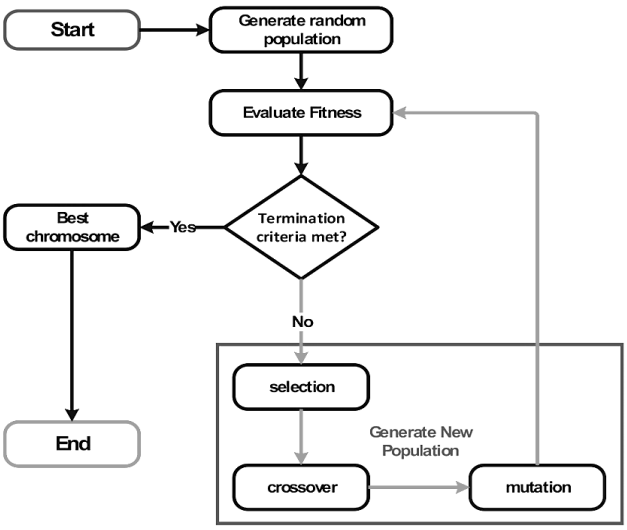
To address these limitations, machine learning has emerged as a promising solution, with significant interest in integrating advanced learning techniques into Intrusion Detection Systems (IDS). Deep learning algorithms have indicated exceptional capabilities across different fields, including speech recognition, natural language processing, and image analysis. Among these, Probabilistic Neural Networks (PNNs) offer several key advantages:

- **Pattern Recognition:** PNNs excel at identifying complex patterns in data, making them well-suited for detecting brute force attacks.
- **Adaptability:** They can learn from new data and dynamically adjust to evolving attack strategies, enhancing their resilience against emerging threats.

This study proposes an IDS framework that leverages PNNs to enhance the detection of brute force attacks. The proposed model aims to improve detection accuracy while minimising false positives, providing organisations with a scalable and effective cybersecurity solution.



[Fig.2: Signature-Based Detection Flow [4]]



[Fig.3: Anomaly-Based Detection Flow [4]]

III. LITERATURE REVIEW

In research [5], deep learning models were employed to enhance intrusion detection systems (IDS). A deep network model with automatic feature extraction was developed to improve the performance of network intrusion detection. The study introduces a novel Intrusion Detection System (IDS) incorporating Gated Recurrent Units (GRU), Recurrent Neural Networks (RNN), Multilayer Perceptrons (MLP), and Softmax modules to address time-dependent intrusions. Experiments were conducted using the KDD 99 and NSL-KDD datasets. The results for BGRU + MLP on the KDD 99 dataset showed an accuracy of 99.89%, a detection rate of 99.42%, and a false positive rate of 0.05%. On the NSL-KDD dataset, BGRU + MLP achieved an accuracy of 99.24%, a detection rate of 99.31%, and a false positive rate of 0.84%. Comparisons with previous studies revealed that BGRU + MLP outperformed earlier methods on both datasets. The study also found that GRU surpassed LSTM, while BGRU outperformed GRU. Additionally, bidirectional RNNs demonstrated superior performance over standard RNNs. Combining RNN and MLP yielded better results than using either model alone. The system was particularly effective in detecting DOS and PROBING attacks, although R2L and U2R attacks remained challenging.

Research [6] addresses a key challenge in IDS: minimizing false alarms. The author developed a descriptive model using different RNN architectures to enhance system reliability. Four RNN models - Bidirectional RNN (BRNN), LSTM, and Bidirectional LSTM (BLSTM) - were tested on the NSL-KDD dataset to detect anomalies in request sequences. These models effectively classified normal and abnormal behaviors and applied this learning to unseen, potentially hazardous requests. The highest recall rates obtained were 98.1% for LSTM in the binary classifier, 87.0% for LSTM in the class-based classifier, and 98.1% in the hidden layer. The findings indicate that Bi-directional LSTM outperformed other RNN models.

Research [7] focuses on improving IDS for detecting brute-force attacks using the CICIDS-2017 dataset. The

© Copyright: All rights reserved.

study employs machine learning algorithms such as Naïve Bayes and J48 decision trees, along with feature selection techniques like Correlation Feature Selection (CFS) and Classifier Subset Evaluator (CSE) to identify relevant classification attributes. The results demonstrated high accuracy (above 99%) with minimal false positives. Discretization methods further enhanced precision and recall, underscoring the importance of advanced feature selection and machine learning in strengthening network security against brute-force attacks.

The widespread use of wireless networks for data transmission has raised significant security and privacy concerns, prompting the development of Intrusion Detection Systems (IDS) as a preventive measure. Despite their importance, IDS performance remains a key issue. Expanding the feature space has proven beneficial for enhancing accuracy in machine-learning-based intrusion detection systems (IDSs). Research [8] presents a deep learning-based IDS utilizing Fully Connected Deep Neural Networks (FFDNNs) combined with a feature selection method. The study evaluates FFDNN-IDS using the NSL-KDD dataset. It compares its performance with that of traditional machine learning models, including Support Vector Machines (SVM), Decision Trees (DT), k-Nearest Neighbours (KNN), and Naïve Bayes. Experimental results indicate that FFDNN-IDS outperforms these methods in accuracy.

Research [9] addresses the complexities of advanced cybersecurity attacks through an intelligent network intrusion detection system leveraging deep learning techniques. The authors implemented Convolutional Neural Networks (CNNs) alongside two recurrent neural network (RNN) variants—Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTMs). Using the NSL-KDD dataset, the training and testing phases were distributed across two workstations: one running Ubuntu 18.04 for CNN implementation and another running macOS 10.14.4 for RNN-LSTM and RNN-GRU. Experimental results demonstrated F1-scores of 98.48% for CNN, 89.54% for RNN-LSTM, and 65.53% for RNN-GRU. CNN achieved the highest accuracy (97.01%), followed by RNN-LSTM (81.60%) and RNN-GRU (50.25%). CNN also outperformed both RNN models in terms of precision and required fewer training epochs. However, recall rates for RNN-LSTM (99.74%) and RNN-GRU (99.88%) surpassed CNN's recall (97.01%).

In research [10], the authors enhanced DDoS detection accuracy by incorporating deep learning techniques into a hybrid IDS. A CNN-based model was proposed, featuring four convolutional layers, two pooling layers, three dropout layers, two dense layers, an attenuation layer, and a softmax layer. Deep learning enabled efficient feature extraction through multiple hidden layers. The model was tested on the NSL-KDD and ISCX IDS 2012 datasets. On NSL-KDD, the CNN achieved a detection rate of 99.94%, an accuracy of 99.90%, and a false positive rate of 0.0004%. These findings highlight the significant improvement deep learning brings to intrusion detection accuracy.

Timely and accurate alerting is crucial to the effectiveness of IDS. To address this challenge, research [11] proposed a machine learning approach using deep learning for intrusion detection. The model was evaluated on NSL-KDD and CIC-IDS2017 datasets. A binary classification approach combined Spark-based k-means clustering and Random Forest (RF), while multi-target anomaly classification incorporated CNN and LSTM. The NSL-KDD dataset achieved a detection rate of 98.57% and an accuracy of 92.90%, while CIC-IDS2017 reached a 99.67% detection rate and 99.87% accuracy. These results demonstrate the effectiveness of distributed machine learning in improving IDS accuracy while reducing false alarms.

Research [12] developed an IDS using a hybrid deep learning strategy that integrates GRUs with enhanced LSTMs to detect abnormal network traffic in cloud computing environments. The CICIDS2018 dataset was used, and the CuLSTMGRU model was designed with Pearson correlation for feature selection. The proposed model achieved 99.76% accuracy, demonstrating improved efficiency and accuracy in cloud-based intrusion detection systems (IDS).

Research [13] explores early detection of network intrusions using deep learning, proposing an end-to-end IDS with CNN. The study leverages the CICIDS2017 dataset to extract key features from raw network traffic automatically. The CNN model achieved an accuracy of 80.3%, emphasizing the potential of deep learning in proactive threat detection.

Research [14] introduces an RNN-based IDS framework incorporating LSTM, GRU, and simple RNN models. The study highlights the growing cybersecurity risks associated with cloud computing and the Internet of Things (IoT). Using XGBoost for feature selection, the framework was tested on NSL-KDD and UNSW-NB15 datasets, achieving test accuracies of 88.13% (XGBoost-LSTM on NSL-KDD) and 87.07% (XGBoost-Simple RNN on UNSW-NB15). These results validate the framework's effectiveness in improving intrusion detection accuracy.

Intrusion detection plays a crucial role in network security, particularly in detecting zero-day attacks. Research [15] addresses feature learning challenges by integrating CNN and GRU in various configurations. Simulations using the CICIDS-2017 dataset resulted in an accuracy of 98.73% and an FPR rate of 0.075%, demonstrating significant improvements in attack detection.

Research [16] presents a neural network-based IDS for detecting brute-force attacks on FTP and SSH protocols. Using the CSE-CIC-IDS2018 dataset, the model was trained on Google Colab, incorporating multiple hidden layers with ReLU activation and a softmax classifier. The model achieved 99.9% accuracy, 98.3% F1-score, 100% precision, and 98% recall, effectively mimicking real-world attack scenarios.

Research [17] focuses on detecting brute-force attacks on SSH and FTP using machine learning. The study evaluates various classifiers, with Random Forest achieving the highest accuracy (99.9%). The research also categorizes brute-force techniques and suggests mitigation strategies such as account lockout policies and two-factor authentication.

Research [18] enhances IDS capabilities in MQTT environments to detect DoS and brute-force attacks. Using the MQTTset dataset, the study applies feature selection techniques (K-Best, PCC, PCA) and evaluates machine

© Copyright: All rights reserved.

learning models (RF, DT, KNN, XGBoost) through ensemble methods. Stacking and voting classifiers achieved 95.38% accuracy, demonstrating the effectiveness of ensemble learning in securing IoT networks.

Reviewing prior studies confirms the effectiveness of machine learning (ML) and deep learning (DL) in cyberattack detection. Consequently, this study proposes a Probabilistic Neural Network (PNN) model for detecting brute-force attacks, one of the most prevalent cybersecurity threats.

Table 1. Comparison Between Previous Related Work

Ref	Dataset	Method	Accuracy
[5]	KDD 99 and NSL-KDD	BGRU + MLP RNN	BGRU + MLP on KDD 99 99.89% BGRU + MLP on NSL-KDD 99.24%
[7]	CICIDS-2017	NB, DT, J48	Reaches over 99%
[8]	NSL-KDD	FFDNN-IDS	Reach 99.54% in the evaluation of multiclass classification
[9]	NSL-KDD	CNN RNN(GRU, LSTM)	CNN 97.01% RNN-LSTM 81.60% RNN-GRU50.25%
[10]	NSL-KDD and ISCXIDS 2012	CNN	99.90 %
[11]	NSL-KDD and the CIC-IDS2017	CNN+ LSTM	NSL-KDD 92.90% CIC-IDS2017 up to 99.87%.
[12]	CSE-CIC-IDS2018	CuLSTMGRU	99.76%
[13]	CICIDS2017	CNN	Up to 98.67%
[15]	CICIDS-2017	CNN–GRU	98.73%
[16]	CSE-CIC-IDS2018	ANN	98.3%
[17]	CSE-CIC-IDS2018	RF	99.9%
[18]	MQTTset	RF, DT, KNN	Reaches 95.38%

IV. RESEARCH METHODOLOGY

This methodology outlines a structured approach to addressing the research problem and achieving its objectives. It ensures that each stage of the research is conducted systematically and rigorously, leading to reliable and meaningful insights. The key steps are as follows:

A. Identifying the Research Topic and Problem: The initial step involves selecting a relevant and compelling research topic and clearly defining the research problem to be investigated.

B. Reviewing Related Literature: A comprehensive review of existing studies is conducted to analyze and compare previous findings. This step helps establish a strong foundation for the research, identifying gaps and areas for further exploration.

C. Developing a Description of the Proposed Model: Insights gained from the literature review inform the creation of a detailed description of the proposed model, including its design, structure, and objectives.

D. Implementing the Proposed Model: The model is developed, tested, and observed to assess its performance in real-world scenarios.

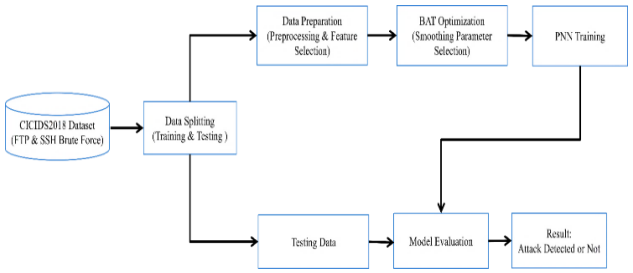
E. Analyzing and Evaluating Results: The final step involves assessing the model's effectiveness, determining its accuracy, and evaluating its ability to address the identified research problem.

This methodology ensures a logical progression from problem definition to model evaluation, providing a robust framework for effectively achieving the research objectives.

V. PROPOSED MODEL

The proposed approach leverages probabilistic neural networks (PNN) to maximize detection accuracy. It encompasses essential stages, including data collection, preprocessing, and model evaluation. Fig.4 illustrates the methodological workflow, detailing each stage of the approach.

This structured approach ensures the efficient identification of brute-force attacks while optimizing detection accuracy.



[Fig.4: The proposed Model Architecture]

A. Data Collection

The CICIDS2018 dataset, a publicly available resource, is utilized in the proposed model. This dataset includes seven types of cyberattacks: brute force attacks, denial-of-service (DoS) attack variants, distributed denial-of-service (DDoS) techniques, DDoS LOIC-HTTP, advanced web application attacks, infiltration tactics, and botnet detection. Table II presents an overview of the dataset, detailing the number of benign and malicious instances.

The proposed model focuses explicitly on FTP and SSH brute-force attacks. To collect benign background traffic, the B-profile technique was applied, incorporating 25 user profiles based on HTTPS, HTTP, FTP, SSH, and email protocols. The dataset consists of five days of network traffic collection, followed by one day of regular activity traffic and two days of injected attacks. These injected attacks include Web Attack, Brute Force FTP, Brute Force SSH, DDoS, DoS, Infiltration, Botnet, and Heartbleed.

The dataset contains 83 feature types, along with two columns for FlowID and one for the label, which provide insights into the forward and backward directions of network flow and packet transmission.

Table 2. The CSE - CIC - IDS 2018 Dataset

Attack Type	Benign	Malicious
Brute Force	667,626	FTP Brute Force: 193,360 SSH Brute Force: 187,589
DoS Attack Types	996,077	DoS Goldeneye: 41,508 DoS Slowloris: 10,990
DDoS LOIC-HTTP	7,372,557	576,191

© Copyright: All rights reserved.

Advanced DDoS Techniques	686,012	DDOS attack-LOIC-UDP: 360,833 DDOS attack-HOIC: 1730
Web Application Attacks	1,048,213	Brute Force-Web: 249 Brute Force-XSS: 79 SQL injection: 34
Infiltration Tactics	1,048,009	151
Botnet Detection	762,384	286,191

B. Preprocessing

Preprocessing is the next step after data collection and is crucial in developing a machine learning-based model. The process begins with data cleaning, where entries with missing or erroneous values are removed. This is followed by **data normalisation, which scales numerical features to the [0, 1] range, thereby** improving the model's convergence and processing efficiency.

C. Feature Selection

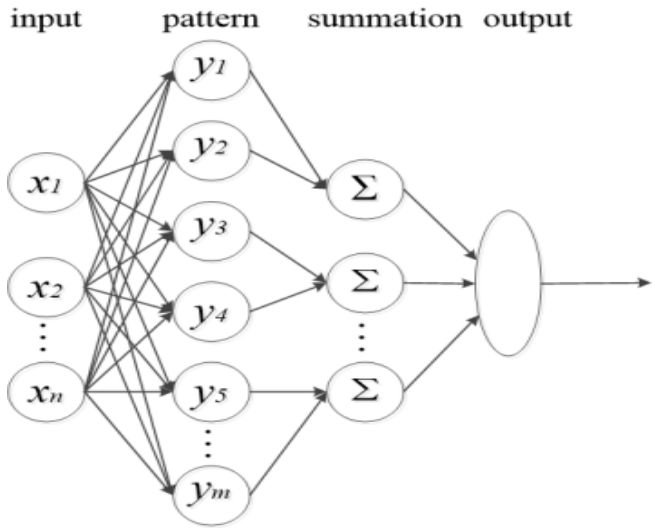
Feature selection is the process of identifying the most relevant features for learning while removing unnecessary or redundant ones. The primary goal is to prevent overfitting and enhance further analysis. In this research, 77 features were retained, while irrelevant features were eliminated to optimize the model's performance.

D. Model Design & Training

The dataset is split into two parts: 80% for training and 20% for testing, ensuring the model's performance is evaluated on unseen data.

E. Probabilistic Neural Network

Probabilistic Neural Networks (PNNs) are employed for classification and regression tasks, leveraging Bayesian inference. By utilizing a probabilistic approach, PNNs integrate data from multiple variables to predict the value of a target variable [19]. The kernel function maps an input to a different dimension (a feature map) and computes the dot products with each feature map for a given input. PNNs offer advantages such as faster training, higher accuracy, and relative insensitivity to noise [20]. Given these benefits, we extend the use of PNNs for detecting brute-force attacks. The hierarchical structure of PNNs consists of four layers: input, pattern, summation, and output. Figure 5 illustrates the fundamental architecture of PNN.



[Fig.5: The PNN Structure [21]]

F. Smoothing Parameter

The smoothing parameter (σ) is critical in determining the model's performance and its ability to generalize from training data to unseen data. A small σ value makes the model highly sensitive to individual training samples, increasing the risk of overfitting. Conversely, a larger σ value results in a smoother, more generalised model, which reduces overfitting but potentially leads to underfitting. Therefore, selecting an optimal σ value is essential for enhancing performance.

In this research, we employ the BAT algorithm, a nature-inspired optimization technique introduced by [22]. This algorithm efficiently finds near-optimal solutions within a reasonable time frame by balancing exploration and exploitation, allowing it to navigate the solution space while avoiding local optima. The BAT algorithm is favoured for its simplicity, flexibility, and minimal parameter tuning requirements, making it well-suited for solving complex optimisation problems. Additionally, it performs well in dynamic and noisy environments, similar to real-world challenges. With its robustness, adaptability, and efficiency, the BAT algorithm serves as a powerful tool for optimization tasks.

G. Model Evaluation

In this research, the model is assessed using a confusion matrix, a powerful statistical tool for evaluating the performance of binary classifiers. It allows for a detailed comparison between predicted and actual outcomes, providing insight into how well the model performs on a held-out dataset. If the model is underperforming or difficult to interpret, it will likely show poor results on the test set. Identifying such issues is most effectively done through a comparative analysis of the test set predictions against the actual values. Upon reviewing the test set, we observed that our classifier produced some false positives and false negatives. With this information, we can retrain the model or make adjustments to improve its performance. Typically, confusion matrices are displayed in a 2x2 grid, where the first row represents the "true state" of the data and the second row shows the "predicted state" from the classifier. Based on the confusion matrix, the model was evaluated using metrics such as accuracy, precision, recall, and F1-Score, as defined in Equations (1)-(4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots \quad (1)$$

$$Precsion = \frac{TP}{TP + FP} \quad \ldots \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad \ldots \quad (3)$$

$$F - Measure = 2 \times \frac{Precsion * Recall}{Precsion + Recall} \quad \ldots \quad (4)$$

Here, TP = true positive, TN = true negative, FP = false positive, and FN = false negative.

© Copyright: All rights reserved.

VI. REQUIREMENT ANALYSIS

This section describes the hardware and software requirements for detecting brute force attacks with the CSE-CIC-IDS2018 dataset. The primary goal of this research is to examine and classify brute force attack patterns related to SSH and FTP using a PNN.

A. Google Colab GPU

The primary hardware used for this project is Google Colab, which provides free access to GPU resources. The system specifications are as follows:

- **GPU Type**: NVIDIA T4
- **GPU Memory**: Up to 16 GB of RAM
- **System RAM**: Approximately 12 GB of virtual machine memory, adequate for data preprocessing and model training tasks.

Utilizing Google Colab offers several advantages:

- The GPU's parallel processing power significantly reduces training times for deep learning models.
- It allows for the handling of larger datasets and more complex models, which would be challenging to run on local machines with limited resources.

B. Programming Environment

The software environment for this research is primarily built on Python, utilising libraries and frameworks that facilitate data manipulation, model development, and evaluation. The key software requirements are as follows:

- **Python**: Version 3.10.12
- **Libraries**: we used the following:
- **NumPy**: Essential for numerical computations, managing multidimensional arrays, and providing core support for data operations.
- **Pandas**: Offers a robust DataFrame structure for efficient data manipulation and analysis, beneficial for handling the CSE-CIC-IDS2018 dataset.
- **Matplotlib & Plotly**: Used for data visualization, plotting, and displaying model performance.
- **Scikit-learn**: Provides machine learning utilities, including tools for model training, evaluation, and preprocessing (e.g., train-test splitting, metric calculation).
- **TensorFlow/Keras**: A platform for developing and training deep learning models, specifically the Probabilistic Neural Network (PNN) used in this study.
- **Imbalanced-learn**: Offers techniques for addressing class imbalance, such as SMOTE (Synthetic Minority Over-sampling Technique), which is crucial for enhancing model performance.

VII. EXPERIMENT AND RESULT ANALYSIS

The model is implemented using Google Colab's GPU capabilities, along with Python's extensive library ecosystem, which enables efficient data processing, model training, and evaluation. This platform supports research by facilitating the effective development and assessment of attack detection methodologies. The goal of the study is to evaluate how well a probabilistic neural network (PNN) optimized with the Bat Algorithm (BAT) can detect brute-force attacks in network traffic. The experimental setup uses the CICIDS2018 dataset, focusing on binary classification to distinguish between malicious brute-force attempts and benign traffic.

The dataset includes a variety of attack types, including brute-force attacks on SSH and FTP. We reduced the dataset by removing unnecessary features and retaining 77 relevant features—one preprocessing step involved eliminating rows with missing or duplicate values. To address class imbalance, we applied SMOTE (Synthetic Minority Over-sampling Technique) to oversample the minority class (malicious traffic), ensuring a balanced dataset. We also employed two feature scaling techniques—standardisation and min-max scaling—which are beneficial for models such as neural networks. To optimize prediction accuracy, we employed the BAT technique to fine-tune smoothing parameters. The performance metrics for each class are provided in Table III.

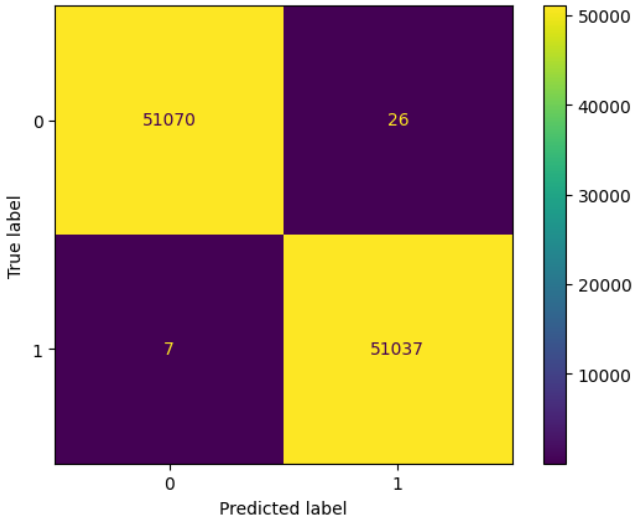
Table 3. Results of Binary Classification

	Precision	Recall	F1-score	Support
Benign	0.99986	0.99949	0.99968	51096
Malicious	0.99949	0.99986	0.99968	51044
Accuracy	___	___	0.99968	102140
Macro avg	0.99968	0.99968	0.99968	102140
Weighted avg	0.99968	0.99968	0.99968	102140

The confusion matrix Fig. 6) shows the performance of the binary classification model:

- 51,070 true negatives (TN) were accurately predicted as 0.
- 26 false positives (FP) were incorrectly predicted as 1.
- Seven false negatives (FN) were incorrectly predicted as 0.
- 51,037 true positives (TP) were accurately predicted as 1.

The confusion matrix shows that the model achieves high accuracy with minimal errors, indicating its robustness and reliability for the task. Additionally, the ROC curve reinforces this finding, demonstrating near-perfect performance with an AUC score of 0.99995.

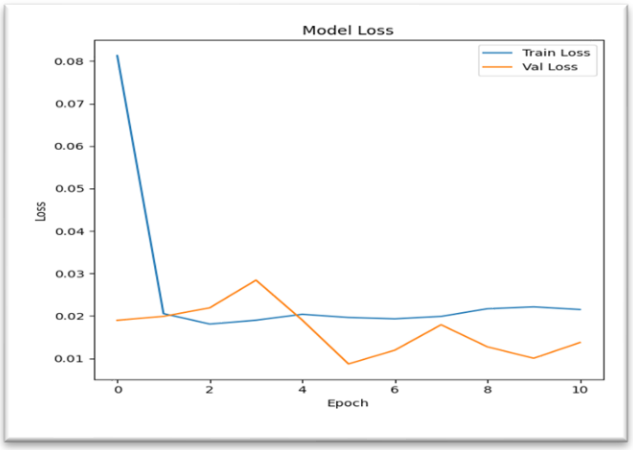


[Fig.6: Confusion Matrix of The Proposed Model]

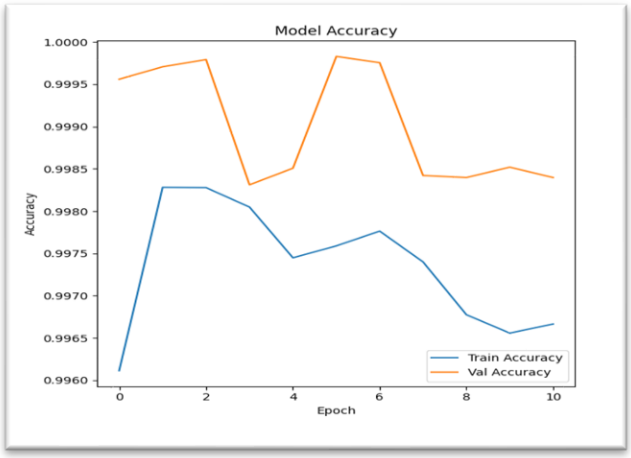
Additionally, Fig. 7 displays two plots: Fig. 7a for model loss and Fig. 7 b for model accuracy. The model loss plot shows the training and validation loss over the epochs, with the

© Copyright: All rights reserved.

training loss (represented by the blue line in Fig. 7a) starting high, rapidly decreasing at the beginning, and stabilising towards the end, indicating that the model's performance on the training data is improving. Conversely, the training accuracy (blue line in Fig. 7 b begins lower and progressively increases with each epoch.



[Fig.7: a. The Loss of the Proposed Model]



[Fig.7: b. The Accuracy of the Proposed Model]

The smoothing parameter sigma was optimised using the Bat Algorithm, a heuristic optimisation technique. The Bat Algorithm successfully determined the optimal sigma value, which was found to be 0.01. This parameter fine-tuned the PNN's regularisation strength, minimising overfitting and enhancing the model's ability to generalise. The final validation loss of 0.0127 reflects the model's strong generalization ability. The proposed model achieved outstanding results with an accuracy of 99.968%, precision of 99.949%, recall of 99.986%, and an F1 score of 99.968%. Table V compares the performance of our proposed model with other models in the literature in terms of accuracy, precision, recall, and F1 score, as shown in Fig. 8.

Table 4. Comparison Between the Proposed Model and Other Models

Reference	Methods	Accuracy	Precision	Recall	F1-Score
[12]	Cu-LSTMGRU	99.76%	99%	99.6%	99.3%
[23]	TCN-LSTM	97.77%	97.94%	97.53%	97.73%
[16]	ANN	99.90%	100%	98%	98.3%
Proposed model	PNN	99.968%	99.949%	99.986%	99.968%

Our model outperforms existing methods in brute-force attack detection, achieving superior accuracy, recall, and F1 score. It is highly effective in correctly identifying true positives while minimizing false positives, resulting in significantly higher accuracy and robustness.

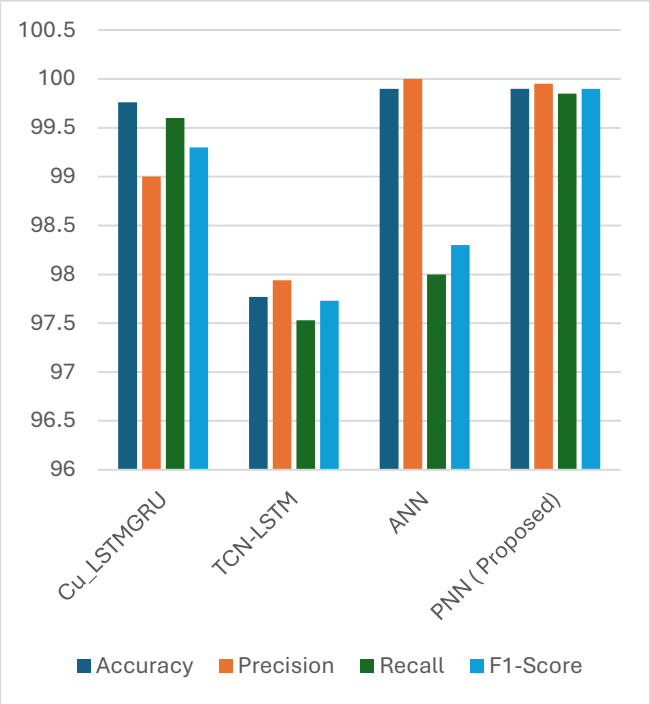


Fig 8. Comparison Between Models]

VIII. CONCLUSION AND FUTURE WORK

This paper presents a model for detecting brute-force attacks using a probabilistic neural network (PNN). The proposed model outperforms existing methods, as demonstrated by the results. We utilised the CICIDS2018 dataset to train the PNN, leveraging the computational power provided by Google Colab. The model effectively identifies SSH/FTP brute-force attacks, achieving remarkable accuracy. The performance metrics of the trained model include an accuracy of 99.968%, a precision of 99.949%, a recall of 99.986%, and an F1-score of 99.968%. Future work will aim to enhance the PNN model by expanding the dataset to incorporate a wider range of attack scenarios and traffic patterns, including more complex threats.

DECLARATION STATEMENT

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 funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Kolekar, V. K., & Vaidya, M. B. (2016). Click and session-based - Captcha as a graphical password authentication scheme for smartphones and web. Proceedings - IEEE International Conference on Information Processing, ICIP 2015. https://doi.org/10.1109/INFOP.2015.7489467
2. Liao, H. J., Richard Lin, C. H., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. In Journal of Network and Computer Applications (Vol. 36, Issue 1). https://doi.org/10.1016/j.jnca.2012.09.004
3. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1). https://doi.org/10.1002/ett.4150
4. Aldallal, A., & Alisa, F. (2021). An effective intrusion detection system to secure data in the cloud using machine learning. Symmetry, 13(12). https://doi.org/10.3390/sym13122306
5. Xu, C., Shen, J., Du, X., & Zhang, F. (2018). An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units. IEEE Access, 6. https://doi.org/10.1109/ACCESS.2018.2867564
6. Elsherif, A. (2018). Automatic Intrusion Detection System Using Deep Recurrent Neural Network Paradigm. Journal of Information Security and Cybercrimes Research. https://doi.org/10.26735/16587790.2018.003.
7. Panwar, S. S., Negi, P. S., Panwar, L. S., & Raiwani, Y. P. (2019). Implementation of machine learning algorithms on the CICIDS-2017 dataset for intrusion detection using WEKA. International Journal of Recent Technology and Engineering, 8(3), 2195–2207. https://doi.org/10.35940/ijrte.C4587.098319
8. Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter-based feature engineering for a wireless intrusion detection system. IEEE Access, 7. https://doi.org/10.1109/ACCESS.2019.2905633
9. Al-Emadi, S., Al-Mohannadi, A., & Al-Senaid, F. (2020). Using Deep Learning Techniques for Network Intrusion Detection. 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020. https://doi.org/10.1109/ICIoT48696.2020.9089524
10. Saraeian, S., & Golchi, M. M. (2020). Application of Deep Learning Technique in an Intrusion Detection System. International Journal of Computational Intelligence and Applications, 19(2). https://doi.org/10.1142/S1469026820500169
11. Liu, C., Gu, Z., & Wang, J. (2021). A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning. IEEE Access, 9. https://doi.org/10.1109/ACCESS.2021.3082147
12. Aldallal, A. (2022). Toward an Efficient Intrusion Detection System Using a Hybrid Deep Learning Approach. Symmetry, 14(9). https://doi.org/10.3390/sym14091916
13. Ahmad, T., Truscan, D., Vain, J., & Porres, I. (2022). Early Detection of Network Attacks Using Deep Learning. Proceedings - 2022 IEEE 14th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2022. https://doi.org/10.1109/ICSTW55395.2022.00020
14. Kasongo, S. M. (2023). A deep learning technique for intrusion detection systems using a Recurrent Neural Network-based framework. Computer Communications, 199. https://doi.org/10.1016/j.comcom.2022.12.010
15. Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., Sharma, B., & Chowdhury, S. (2023). Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. Sensors, 23(2). https://doi.org/10.3390/s23020890
16. Alotibi, N., & Alshammari, M. (2023). Deep Learning-based Intrusion Detection: A Novel Approach for Identifying Brute-Force Attacks on FTP and SSH Protocols. International Journal of Advanced Computer Science and Applications, 14(6). https://doi.org/10.14569/IJACSA.2023.0140612
17. Hamza, A. A., & Al-Janabi, R. J. surayh. (2024). Detecting Brute Force Attacks Using Machine Learning. BIO Web of Conferences, 97. https://doi.org/10.1051/bioconf/20249700045
18. Hanif, A. Al, & Ilyas, M. (2024). Enhance the Detection of DoS and Brute Force Attacks within the MQTT Environment Through Feature Engineering and Employing an Ensemble Technique. International Journal of Artificial Intelligence & Applications, 15(4), 01–19. https://doi.org/10.5121/ijaia.2024.15401
19. Chang, D. T. (2021). Probabilistic Deep Learning with Probabilistic Neural Networks and Deep Probabilistic Models. https://doi.org/10.48550/arXiv.2106.00120
20. Alarfaj, F. K., & Khan, N. A. (2023). Enhancing the Performance of SQL Injection Attack Detection through Probabilistic Neural Networks. Applied Sciences (Switzerland), 13(7). https://doi.org/10.3390/app13074365
21. Zhao, G., Zhang, C., & Zheng, L. (2017). Intrusion detection using deep belief network and probabilistic neural network. Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, 1. https://doi.org/10.1109/CSE-EUC.2017.119
22. Yang, X. S. (2010). A new metaheuristic Bat-inspired Algorithm. Studies in Computational Intelligence, 284. https://doi.org/10.1007/978-3-642-12538-6_6
23. Mezina, A., Burget, R., & Travieso-Gonzalez, C. M. (2021). Network Anomaly Detection with Temporal Convolutional Network and U-Net Model. IEEE Access, 9. https://doi.org/10.1109/ACCESS.2021.3121998

AUTHOR'S PROFILE

Hanadi Hassan Alosimy received a Computer Science B.Sc. degree in 2020 from Taif University. Currently, she is pursuing a Master of Science (M.Sc.) in Cybersecurity at Taif University in Taif, Saudi Arabia. Her academic journey has profoundly shaped her interest in safeguarding digital systems against the ever-evolving landscape of cyber threats. Through rigorous study and practical experiences, she has developed a strong passion for leveraging advanced technologies to tackle real-world challenges in cybersecurity. This includes exploring innovative approaches such as artificial intelligence, machine learning, and cryptographic techniques to enhance system resilience, protect sensitive information, and preempt potential vulnerabilities. Her commitment to this field is driven by the critical need to Secure digital infrastructures in an increasingly interconnected and technology-dependent world.

Jawaher Talak Alzaidi received a B.Sc. degree in 2019 in Information Technology from the Computers and Information Technology College at Taif University and is currently studying for an M.Sc. in Cybersecurity at Taif University, Taif, Saudi Arabia. Holds a basic cybersecurity certificate from Cisco Networking Academy, showcasing foundational expertise in cybersecurity principles and practices. She has practical experience in digital forensics investigations, where she has worked on analyzing and recovering data to uncover cyber incidents, as well as in designing and implementing intrusion detection systems (IDS) and intrusion prevention systems (IPS) to safeguard digital environments from potential threats. Her academic and professional interests extend to cutting-edge technologies, including Artificial Intelligence, Machine Learning, and Deep Learning. She aims to leverage these technologies to advance and optimize Intrusion Detection Systems, contributing to the development of more intelligent and adaptive cybersecurity solutions.

© Copyright: All rights reserved.

Samah Hazzaa Alajmani received her B.Sc. degree in 2004 and Ph.D. degree in 2019 from 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, Artificial Intelligence (AI), Internet of Things (IoT), Deep Learning (DL) and Machine learning (ML). Ben Soh, an Associate Professor, obtained his PhD in Computer Science & Engineering (in the area of Secure and Fault-Tolerant Computing under the tutelage of Prof. TS Dillon) from La Trobe. Since then, he has successfully supervised to completion 14 PhD students and published more than 200 peer-reviewed research papers. He has made significant contributions in the following research areas: Fault-Tolerant and Secure Computing, Cloud Computing, Information Systems Research, Pervasive Wireless Network Communications, Educational Technology and Business Process Management. Currently, he serves as La Trobe's cybersecurity programs advisor and coordinator.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.