Deep Learning-based Intrusion Detection: A Novel Approach for Identifying Brute-Force Attacks on FTP and SSH Protocol

Noura Alotibi, Majid Alshammari
Department of Information Technology-College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

***Abstract*—As networks continue to expand rapidly, the number and diversity of cyberattacks are also increasing, posing a significant challenge for organizations worldwide. Consequently, brute-force attacks targeting FTP and SSH protocols have become more prevalent. IDSes offer an essential tool to detect these attacks, providing traffic analysis and system monitoring. Traditional IDSes employ signatures and anomalies to monitor information flow for malicious activity and policy violations; however, they often struggle to effectively identify unknown or novel patterns. In response, we propose a novel intelligent approach based on deep learning to detect brute-force attacks on FTP and SSH protocols. We conducted an extensive literature review and developed a metric to compare our work with existing literature. Our findings indicate that our proposed approach achieves an accuracy of 99.9%, outperforming other comparable solutions in detecting brute-force attacks.**

Keywords—Artificial neural networks; machine learning; deep learning; intrusion detection system; detecting brute force attacks on SSH and FTP protocols

I. Introduction

Over the past decade, network security has emerged as a major research area due to the growing interest and advancements in internet and communication technologies. The security of networks and their connected assets in cyberspace is primarily protected by various technologies, such as firewalls, antivirus software, and Intrusion detection systems (IDSes) [1]. However, as attacks become more sophisticated, non-traditional techniques are required to detect them. Consequently, existing IDSes have proven to be ineffective at detecting a wide range of threats, including zero-day attacks, and at reducing false alarm rates (FARs) [2].

Researchers have investigated the potential application of machine learning (ML), including deep learning, to aid in detecting these attacks. ML aims to extract valuable information from large volumes of data [3] and serves as a powerful approach for gathering useful data from network traffic and predicting normal and abnormal events based on learned patterns. Machine learning models rely heavily on feature engineering to learn essential information from network traffic [4]. Due to their structure, deep learning models do not require feature engineering and are capable of automatically learning complex features from raw data [5].

Although deep learning models are still in their early stages, there is considerable potential for advancing this technology. Recently, researchers have begun proposing deep learning models to improve the effectiveness of attack identification. Among the most well-known attacks on networks are brute-force attacks on Secure Shell (SSH) protocol and File Transfer Protocol (FTP). The SSH protocol is used for secure remote login over an insecure network [6], while the FTP protocol is employed to transfer data between a client and a server [4].

Therefore, in this paper, we propose a deep learning-based model for detecting brute-force attacks on FTP and SSH protocols. Brute-force attacks targeting these protocols have become increasingly significant security risks to organizations. By leveraging the capabilities of deep learning, our model aims to overcome the limitations of traditional IDSes and improve the accuracy and efficiency of brute-force attack detection on FTP and SSH protocols. Through an extensive evaluation and comparison with existing literature, our findings demonstrate that the proposed model achieves an accuracy of 99.9%, outperforming other comparable solutions in detecting brute-force attacks.

The remainder of this paper is organized as follows: Section II provides a review of the literature. Section III presents the research methodology. Section IV discusses the proposed model. Section V presents the findings and analysis of the proposed model. Finally, Section VI concludes the paper.

II. Related Works

Cybersecurity is a vast field designed to combat various attack pathways [7]. Since 2009, research has focused on utilizing artificial neural networks (ANNs) to improve anomaly detection and IDS identification. As a result, the application of ANNs in IDS and malware detection is still a relatively new concept [8]. Previous researchers have attempted to address issues related to overfitting, excessive memory utilization, and high overhead associated with traditional IDS detection. A two-layer, feed-forward ANN was suggested for this purpose. According to the authors, their method produced outcomes similar to traditional procedures but required less computational effort. This technique was tested using the benchmark dataset KDD'99. Since the machine requires time to process the data, the paper concluded that having fewer data points was preferable [9]. Researchers evaluated pruning an ANN as part of network optimization, which involved removing neural nodes from the input or hidden layers. As a result, their ANN became faster due to fewer computations needing to be processed. When tested as an IDS, the ANN in [9] showed promising performance.

Rather than delivering the inputs directly from the dataset, [10] used principal component analysis (PCA) as a feature representation before feeding the data to the ANN. As the paper illustrated, this reduced the memory resources needed and the amount of training time required. In terms of accuracy, the two tested approaches produced similar results, making PCA the better alternative. Although using a kernel PCA improves ANN training time, it consumes significantly more memory than standard PCA, which is a drawback. Since the accuracy measures for both techniques are similar, the authors of [11] discussed the use of a combination of techniques as a superior alternative. Because GPUs are well-suited for ANN computations, research has explored using them to accelerate ANN-based IDSes. A performance boost has been demonstrated in [12]. The authors of [13] compared a support vector machine (SVM), Naïve Bayes (NB), and a C4.5 classifier to an ANN with one hidden layer. The ANN yielded equivalent or better results in attack detection than the other algorithms evaluated by the paper, and it did so with fewer computations due to the simplified structure of a three-layered ANN model. The tests were conducted on the NSL-KDD database, which replaced KDD'99 as the current benchmark.

Artificial intelligence-based IDS models continue to face two major issues. Traditional machine learning (ML) algorithms are generally fast but have a high false positive rate (FPR). Deep learning, on the other hand, offers excellent accuracy and low FPR but requires significant computation time. As a result, the authors of [14] proposed a solution that provided the best of both worlds. The suggested solution was an OS-based monitoring algorithm that used standard ML as a quick monitoring device, referred to as the standard stage; when a classification falls into the borderline state, the method's second stage is initiated. The second stage, dubbed uncertain, employs deep learning as the final decision-maker on a process. The authors of [15] examined malware detection flow based on a convolutional neural network (CNN). They found that detection approaches were overly reliant on specific packet elements, such as the port number, which created a blind spot in security, as some malware uses unpredictable port numbers and protocols. Instead of these packet elements, they proposed 35 features derived from the Stratosphere IPS project's data. To address the data balance problem, 2000 data points were selected in each class. Nestmate was used to extract 35-flow static features that were then fed into a CNN and three different machine learning methods for evaluation, including SVM, random forest (RF), and multi-layer perceptron (MLP). Data from the Stratosphere IPS project was used to train the models, as it is publicly available. The CNN architecture consisted of one input layer, five feature map layers, a flatten layer, two hidden layers, and one output layer. The authors concluded that the RF algorithm outperformed other approaches on all three examined indicators: accuracy, specificity, and sensitivity [16].

Table I and Table II display supervised algorithms, as well as autoencoder and deep belief network architectures used for intrusion detection, respectively.

TABLE I. COMPARISON OF INTRUSION DETECTION SCHEMES USING ANN AND CNN ARCHITECTURES

Scheme	Data used	Model architecture	Result in %
[17]	CICIDS 2017, UNSW-NB15, NSL-KDD, Kyoto, WSN-DS	ANN + ReLU activation	Accuracy: 78.5, 95.6, Precision: 81.0, 96.2, Recall: 78.5, 95.6, F1- score: 76.5, 95.7
[18]	ISCX VPN	CNN	accuracy: 99.85
[19]	NSL-KDD	ANN + ReLU activation	Accuracy: 86.35, Precision: 81.86, Recall:77.32, F1-score: 73.89, FAR: 0.1619
[20]	NSL-KDD	ANN	accuracy: 98.27, recall:96.5, FAR: 0.0257
[21]	KDD 99	ANN + ReLU activation	Accuracy: 99.01, Recall:99.81, FAR: 0.0047
[22]	USTC-TFC2016	CNN	Accuracy: 99.17, Precision: 99, Recall:98, F1-score: 98
[23]	Network data Simulated by IoT	ANN + Sigmoid activation	Accuracy: 99

TABLE II. COMPARISON OF INTRUSION DETECTION SCHEMES USING LSTM RNN AND OTHER DEEP LEARNING ARCHITECTURES

Scheme	Data used	Model architecture	Result in %
[8]	NSL-KDD	LSTM	Accuracy: 98.94, Recall:99.23, FAR: 9.86
[24]	ISCX 2012, AWID	embedding + LSTM + sigmoid	Accuracy: 99.91, Precision: 99.85, Recall:99.96
[10]	KDD 99, NSL-KDD	GRU + BGRU	Accuracy: 99.24, Recall:99.31, FAR: 0.84
[12]	Vehicle network data	LSTM	Accuracy: 86.9
[25]	NSL-KDD, binary and 5-class classification	RNN	Accuracy: 81.29
[15]	KDD 99	LSTM network	Accuracy: 97.54, Precision: 97.69, Recall:98.95, FAR: 9.98
[26]	CSE-CIC-IDS2018	Broad Learning System	Accuracy: 97.08 F1- score: 77.89 Precision: NA Recall: NA
[27]	CSE-CIC-IDS2018	LSTM+ SMOTE algorithm	Accuracy :96.2 F1- score: NA Precision :96 Recall :96
[28]	CSE-CIC-IDS2018	Spark ML + Conv-AE	Accuracy: 98.20 F1- score: 98 Precision: NA Recall :98

III. METHODOLOGY

The methodology section outlines the process followed in developing the intrusion detection model using deep learning techniques. The chosen algorithm is based on previous studies, and the model utilizes ANNs with ReLU and SoftMax activation functions. The CSE-CIC-IDS 2018 dataset, specifically the FTP/SSH brute-force attacks, serves as the basis for the model. The entire process is broken down into distinct stages, as detailed below:

1) *Obtain the proposed benchmark dataset*: The CSE-CIC-IDS 2018 dataset is acquired, containing eight different attack types. Only FTP/SSH brute-force attacks are used in this study.

2) *Prepare the data*: Data preprocessing involves correcting issues such as missing values and outliers, ensuring that the dataset is clean and ready for analysis.

3) *Use exploratory analysis*: This step involves understanding the dataset's content and selecting the most suitable algorithm for the given problem.

4) *Train the model*: The best-performing algorithm from the literature review is used to train the model on the prepared dataset.

5) *Evaluate the model*: Evaluation techniques are employed to assess the model's performance and ensure that it meets the desired accuracy and detection standards.

6) *Optimize the model*: If the model's performance is unsatisfactory, alternative algorithms are considered or the current model's parameters are adjusted to improve its effectiveness.

The model is implemented using Google Colab, Python programming language, and the Scikit-learn library. The algorithm implementation is divided into four stages:

Stage 1: Input features are generated from the data preprocessing and representation stages and supplied to the neural network's input layer.

Stage 2: Neural network layers are initialized with random weights, which are used throughout the training phase.

Stage 3: The network accepts the input and begins the training process. The feature identifies the output probabilities by going through the forward propagation phases (dense, ReLU, and operations, as well as the forwarding propagation of hidden layer 3).

Stage 4: The intended output error value is computed and compared to the produced output. Validation is performed after every 50 iterations to assess the model's performance and make adjustments as needed.

This methodology provides a systematic approach to developing an effective intrusion detection model, from obtaining the dataset to training and evaluating the model.

IV. PROPOSED MODEL

To implement the proposed algorithm, it is crucial to obtain the CSE-CIC-IDS2018 benchmark dataset. The data is organized in a CSV file with columns such as FlowID, Destination-IP, Source-Port, and Protocol. The dataset includes over 80 network traffic features representing various attack types, including denial-of-service, Heartbleed, web attacks, botnet, and infiltration attacks.

Brute-force attacks are prevalent against networks as they exploit weak login and password combinations. Our model focuses on identifying SSH and MySQL accounts on the primary server targeted by dictionary brute-force attacks.

The following steps outline the data preprocessing:

1) *Data cleaning and normalization*: Convert all required features to nominal values and clean the data, depending on the created features.

2) *Normalization of data*: Normalize numeric feature values to a chosen scale, such as the [0, 1] range, to decrease data scale and improve model accuracy and processing time.

3) *Splitting data*: Divide the data into three parts: a training set with 60% of the data, a validation set with 20%, and a testing set with the remaining 20%.

Fig. 1 displays the architecture of the proposed model, while Fig. 2 shows the neural network architecture of our multiclass detection model. The ANN aims to minimize the number of information parameters needed by employing equivariant representation, parameter sharing, and sparse interactions. The network consists of multiple hidden layers, as well as an input and output layer.

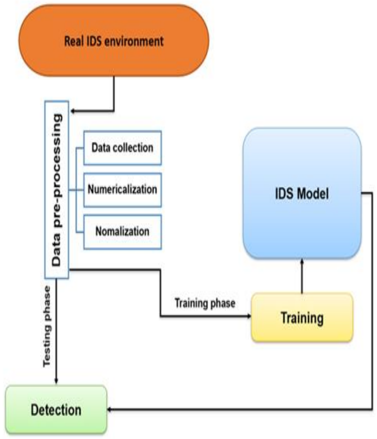

Fig. 1. Architecture of the proposed model.

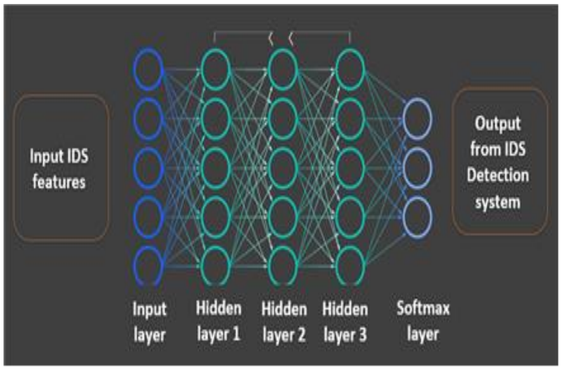

Fig. 2. Proposed neural network architecture of our multiclass detection model.

The model parameters are as follows:

- Input layer: Contains neurons equal to the number of input data features, utilizing the ReLU activation function.
- Hidden layers (1, 2, and 4): Dense layers with a specified number of neurons, also employing the ReLU activation function.
- Output layer: A softmax layer for classification with three output classes.

In this model, we use a softmax classifier for multiclass classification of brute-force attacks targeting SSH and FTP. The softmax layer effectively represents category distributions by normalizing the exponent of output values. It is primarily used in the output layer, providing a differentiable function that reflects the probability of the output.

V. Findings and Analysis

In the Findings and Analysis section, we present the performance of our proposed model and compare it with current research. Our proposed model achieved superior results with over 99.9% accuracy in comparison to the existing literature. Fig. 3, which shows the learning curves during the training process, demonstrates the performance of our proposed model.

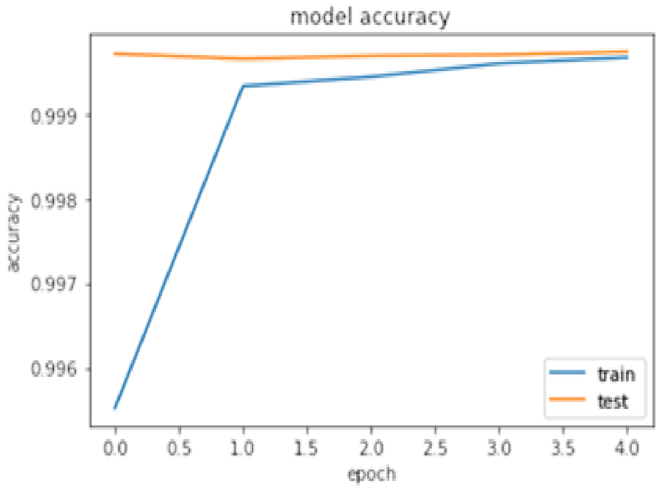

Fig. 3. The accuracy of the proposed model.

Table III compares our proposed model with current research based on accuracy, precision, recall, and F-score. Our model produced superior results, with 99.9% accuracy, 98.3% F-score, 100% precision, and 98% recall.

The results of the proposed model, which employs an Artificial Neural Network (ANN) on the CSE-CIC-IDS2018 dataset, demonstrate a significant improvement in performance compared to existing research. With an accuracy of 99.9%, F1-score of 98.3%, precision of 100%, and recall of 98%, the proposed model outperforms other methods in the literature.

TABLE III. Comparison between the Proposed Model and Current Research

Scheme	Dataset	Model Architecture	Result in %
[26]	CSE-CIC-IDS2018	Broad Learning System	Accuracy: 97.08 F1- score : 77.89 Precision :NA Recall :NA
[27]	CSE-CIC-IDS2018	LSTM+ SMOTE algorithm	Accuracy :96.2 F1- score : NA Precision :96 Recall :96
[28]	CSE-CIC-IDS2018	Spark ML + Conv-AE	Accuracy : 98.20 F1- score : 98 Precision :NA Recall :98
Proposed Model	CSE-CIC-IDS2018	ANN	Accuracy : 99.9 F1- score : 98.3 Precision : 100 Recall : 98

In comparison, the Broad Learning System (BLS) achieved an accuracy of 97.08% and an F1-score of 77.89%, while the model using LSTM with SMOTE algorithm reached an accuracy of 96.2% and a precision and recall of 96%. Finally, the model employing Spark ML with Conv-AE obtained an accuracy of 98.20%, F1-score of 98%, and a recall of 98%. These results show that the proposed ANN model is significantly more accurate than other models, particularly in terms of precision.

The high precision of the proposed model indicates that it is particularly effective at correctly identifying true positives (i.e., correctly detecting attack instances) and minimizing false positives (i.e., misclassifying benign instances as attacks). This is an essential aspect of an intrusion detection system, as it ensures that genuine threats are identified while minimizing the risk of false alarms.

The recall rate of 98% for the proposed model, while slightly lower than its precision, is still noteworthy. This indicates that the model can successfully identify a high proportion of true positive instances from the total number of actual positive instances. In the context of intrusion detection, this means that the proposed model is effective at detecting most of the attacks in the dataset.

VI. Conclusion and Future Work

In this research, we introduced a novel model for detecting network intrusions using a deep neural network. Our proposed model demonstrates superior performance compared to other existing approaches, as evidenced by the results obtained.

We utilized the comprehensive CSE-CIC-IDS2018 dataset to train our powerful neural network model, taking advantage of Google Colab's computational resources. The model effectively defends against SSH/FTP brute-force attacks and is designed to closely emulate real-world scenarios. By employing a hidden real-time test dataset throughout its training and development, the model's performance can be more accurately assessed. Comparison of our model's results with various evaluation metrics reveals its superior ability to detect brute-force attacks, outperforming other recent research studies. The key metrics obtained are 99.9% accuracy, 98.3% F1-score, 100% precision, and 98% recall.

Future work could focus on refining the artificial neural network model and comparing its performance with other machine learning models, such as Support Vector Machines (SVM), logistic regression, decision trees, and random forests. These comparisons would provide valuable insights into the model's performance and potential for further enhancement. Additionally, future research may explore the applicability of the proposed model to a wider range of cyberattacks, contributing to the ongoing development of robust intrusion detection systems.

REFERENCES

[1] S. Wen, Q. Meng, C. Feng, and C. Tang, ‘Protocol vulnerability detection based on network traffic analysis and binary reverse engineering’, PloS one, vol. 12, no. 10, p. e0186188, 2017.

[2] I. Mukhopadhyay, M. Chakraborty, S. Chakrabarti, and T. Chatterjee, ‘Back propagation neural network approach to Intrusion Detection System’, 2011, pp. 303–308.

[3] H. A. Sonawane and T. M. Pattewar, ‘A comparative performance evaluation of intrusion detection based on neural network and PCA’, 2015, pp. 0841–0845.

[4] B. Subba, S. Biswas, and S. Karmakar, ‘A neural network based system for intrusion detection and attack classification’, 2016, pp. 1–6.

[5] T. M. Pattewar and H. A. Sonawane, ‘Neural network based intrusion detection using Bayesian with PCA and KPCA feature extraction’, 2015, pp. 83–88.

[6] Z. Zali, M. R. Hashemi, and H. Saidi, ‘Real-time attack scenario detection via intrusion detection alert correlation’, 2012, pp. 95–102.

[7] M. Yeo et al., ‘Flow-based malware detection using convolutional neural network’, 2018, pp. 910–913.

[8] F. Jiang et al., ‘Deep learning based multi-channel intelligent attack detection for data security’, IEEE transactions on Sustainable Computing, vol. 5, no. 2, pp. 204–212, 2018.

[9] A. Diro and N. Chilamkurti, ‘Leveraging LSTM networks for attack detection in fog-to-things communications’, IEEE Communications Magazine, vol. 56, no. 9, pp. 124–130, 2018.

[10] K. Xu, Y. Li, R. H. Deng, and K. Chen, ‘Deeprefiner: Multi-layer android malware detection system applying deep neural networks’, 2018, pp. 473–487.

[11] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, ‘Network traffic anomaly detection using recurrent neural networks’, arXiv preprint arXiv:1803.10769, 2018.

[12] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, ‘Cloud-based cyber-physical intrusion detection for vehicles using deep learning’, Ieee Access, vol. 6, pp. 3491–3508, 2017.

[13] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, ‘A deep learning approach for network intrusion detection system’, Eai Endorsed Transactions on Security and Safety, vol. 3, no. 9, p. e2, 2016.

[14] L. Bontemps, V. L. Cao, J. McDermott, and N.-A. Le-Khac, ‘Collective anomaly detection based on long short-term memory recurrent neural networks’, 2016, pp. 141–152.

[15] T. H. T. Le, N. H. Tran, P. L. Vo, Z. Han, M. Bennis, and C. S. Hong, ‘Contract-based cache partitioning and pricing mechanism in wireless network slicing’, 2017, pp. 1–6.

[16] A. Taylor, S. Leblanc, and N. Japkowicz, ‘Anomaly detection in automobile control network data with long short-term memory networks’, 2016, pp. 130–139.

[17] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, ‘Deep learning approach for intelligent intrusion detection system’, Ieee Access, vol. 7, pp. 41525–41550, 2019.

[18] Y. Zeng, H. Gu, W. Wei, and Y. Guo, ‘$ Deep-Full-Range $: a deep learning based network encrypted traffic classification and intrusion detection framework’, IEEE Access, vol. 7, pp. 45182–45190, 2019.

[19] W. Wang et al., ‘HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection’, IEEE access, vol. 6, pp. 1792–1806, 2017.

[20] A. A. Diro and N. Chilamkurti, ‘Distributed attack detection scheme using deep learning approach for Internet of Things’, Future Generation Computer Systems, vol. 82, pp. 761–768, 2018.

[21] X. Jin, J. Zhou, H. Dong, W. Lou, J. Wang, and F. Wang, ‘Research on new military plotting system architecture based on AutoCAD secondary development’, 2017, pp. 313–317.

[22] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, ‘Malware traffic classification using convolutional neural network for representation learning’, 2017, pp. 712–717.

[23] E. Hodo et al., ‘Threat analysis of IoT networks using artificial neural network intrusion detection system’, 2016, pp. 1–6.

[24] A. Diro and N. Chilamkurti, “Leveraging LSTM networks for attack detection in fog-to-things communications,” IEEE Commun. Mag., vol. 56, no. 9, pp. 124–130, 2018..

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

[26] A. L. G. Rios, Z. Li, K. Bekshentayeva, and L. Trajković, ‘Detection of denial of service attacks in communication networks’, 2020, pp. 1–5.

[27] P. Lin, K. Ye, and C.-Z. Xu, ‘Dynamic network anomaly detection system by using deep learning techniques’, 2019, pp. 161–176.

[28] ‘M. A. Khan and J. Kim, “Toward developing efficient Conv-AE-based intrusion detection system using heterogeneous dataset,” Electronics, vol. 9, no. 11, p. 1771, 2020.’.