Journal of Integrated
SCIENCE & TECHNOLOGY

BDMDNA : Design of an efficient Blockchain-based Deep learning model for identification and mitigation of dynamic network attacks

Nikita Bahaley,* Avinash Sharma

Madhyanchal Professional University, Ratibad, Bhopal, Madhya Pradesh, India.

Received on: 09-Oct-2023, Accepted and Published on: 03-Jun-2024

Article

ABSTRACT

The increasing number of cyber-attacks on network systems has become a significant challenge for network security under real-time network scenarios. Deep learning models have proven to be effective in identifying network attacks. However, these models require a large amount of data for training, and their implementation can be computationally expensive when deployed on large-scale networks. To overcome these issues, this paper proposes a blockchain-based deep learning model that utilizes the advantages of blockchain to enhance the efficiency and security of network attack identification and mitigation. The proposed model uses a novel Proof-of-Wireless-Trust (PoWT) consensus algorithm to validate and secure the training data, and a customized Binary Cascaded Deep Learning Model (BCDLM) for training the model w.r.t. multiple attack signatures. The blockchain-based model is designed to detect and mitigate dynamic network attacks in real-time, thereby enhancing the security of network systems. The proposed model is evaluated using different network datasets.

Keywords: Blockchain, Security, Cascaded Learning, CNN, Learning Model.

INTRODUCTION

Cyber-attacks have risen as a result of the quick development of technology and the growing reliance on network systems. Because attackers are becoming more skilled and creating new methods to take advantage of weaknesses in network systems, network attacks have grown to be a significant challenge for network security professionals. Effective ways to improve the security of network systems are required, as the detection and mitigation of network attacks have become more important than ever for real-time scenarios.[1-3]

The ability to recognize network attacks using deep learning models has shown promising results due to use of Bayesian Inference (BI).[4-6] These models can recognize patterns and anomalies in network traffic and can identify network attacks by using sophisticated algorithms and vast amounts of data. Deep learning models, however, can be computationally expensive to implement, and the effectiveness of the model depends on the calibre and volume of the training data samples.[7-9] The security of the training data is also a concern because it is vulnerable to manipulation, which could produce false results.

We suggest a deep learning model for blockchain-based dynamic network attack detection and mitigation in this paper. The suggested model makes use of blockchain's benefits to improve the effectiveness and security of network attack detection and mitigation. The model uses a deep learning algorithm to be trained and a consensus algorithm to guarantee the accuracy of the training data. The blockchain-based model offers a practical way to improve the security of network systems by detecting and mitigating dynamic network attacks in real-time scenarios.

The remaining sections of the paper are structured as follows. A review of related work on blockchain and deep learning in network security is presented in Section 2 of this text. The proposed deep learning model for blockchain-based dynamic network attack detection and mitigation is presented in Section 3 of this text. A discussion of the findings is presented in Section 4 of this text, and conclusions and recommendations are presented in Section 5 as conclusive analysis of this text.

*Ms. Nikita S. Bahaley, Research Scholar, Madhyanchal Professional University, Ratibad, Bhopal, Madhya Pradesh, India
Email: nikitabahaley@gmail.com

Cite as: *J. Integr. Sci. Technol., 2024, 12(6), 829.*
URN:NBN:sciencein.jist.2024.v12.829
DOI: 10.62110/sciencein.jist.2024.v12.829

Literature Review of Existing Models used for identification of Network Attacks

In today's linked world, network attacks are becoming a bigger worry as the number of attacks rises yearly. Many academics have created machine learning algorithms to recognize and stop these attacks. We will discuss about the current machine learning algorithms for identifying network attacks in this section.[27,28]

A lot of people use neural networks to detect network attacks. They can recognize trends in real-time communication after being taught on a collection of well-known attack patterns. The Multi-Layer Perceptron is one of the neural network algorithms that is most frequently used (MLP). It has been demonstrated that the MLP is useful for spotting network attacks like DoS, port searches, and infiltration efforts. The drawbacks of neural networks include their high training data requirements and potential for over fitting scenarios which are tackled by Convolutional Neural Networks (CNN), Singular Value Decomposition (SVD) & Long Short-Term Memory Networks with Condition Generative Adversarial Networks (LSTM CGAN).[29-32]

Another machine learning model used for identifying network attacks is decision trees.[33-35] Decisions and potential outcomes are represented using decision trees, which resemble branches. They can recognize trends in real-time communication after being taught on a collection of well-known attack patterns.[36-38] Decision trees are useful for spotting network attacks like port searches, denial-of-service attacks, and infiltration efforts. They can be used to produce guidelines for identifying network attacks because they are reasonably simple to comprehend and analyze for different use cases via Federated Learning (FL).[39]

Network attacks are recognized using machine learning models called Support Vector Machines (SVMs). Network attacks like DoS, port searches, and infiltration efforts can all be detected by SVMs. They are especially helpful for spotting attacks that are challenging for other machine learning algorithms to recognize. SVMs can be used to create algorithms for identifying network attacks because they are reasonably simple to train for different use cases.

As a result, a variety of machine learning models, such as deep learning, neural networks, decision trees, support vector machines, random forests, and support vector machines, are used to identify network attacks. Each of these models has advantages and disadvantages, and the best model to use will rely on the particular requirements of the company. But it is evident that machine learning models are useful for spotting network assaults and are a key component of keeping networks secure.

Proposed Design of an Efficient Blockchain-based Deep Learning Model for Identification & Mitigation of Dynamic Network Attacks

From the review of existing attack detection methods, it can be observed that deep learning models have proven to be effective in identifying network attacks. However, these models require a large amount of data for training, and their implementation can be computationally expensive when deployed on large-scale networks. To overcome these issues, this section proposes design of an efficient blockchain-based deep learning model that utilizes the advantages of blockchain to enhance the efficiency and security of network attack identification and mitigation even under larger network deployments. As per flow of the model in figure 1, it can be observed that the proposed model uses a novel Proof-of-Wireless-Trust (PoWT) consensus algorithm to validate and secure the training data, and a customized Binary Cascaded Deep Learning Model (BCDLM) for training the model w.r.t. multiple attack signatures. The blockchain-based model is designed to detect and mitigate dynamic network attacks in real-time, thereby enhancing the security of network systems.

As per the flow of proposed model, it can be observed that initially a Binary Cascaded Deep Learning Model is used to identify real-time attacks. This model initially performs clustering of data packets into ‘attack’, and ‘non-attack’ groups, which is done via estimation of a novel distance metric between temporal & spatial node performance under different scenarios.

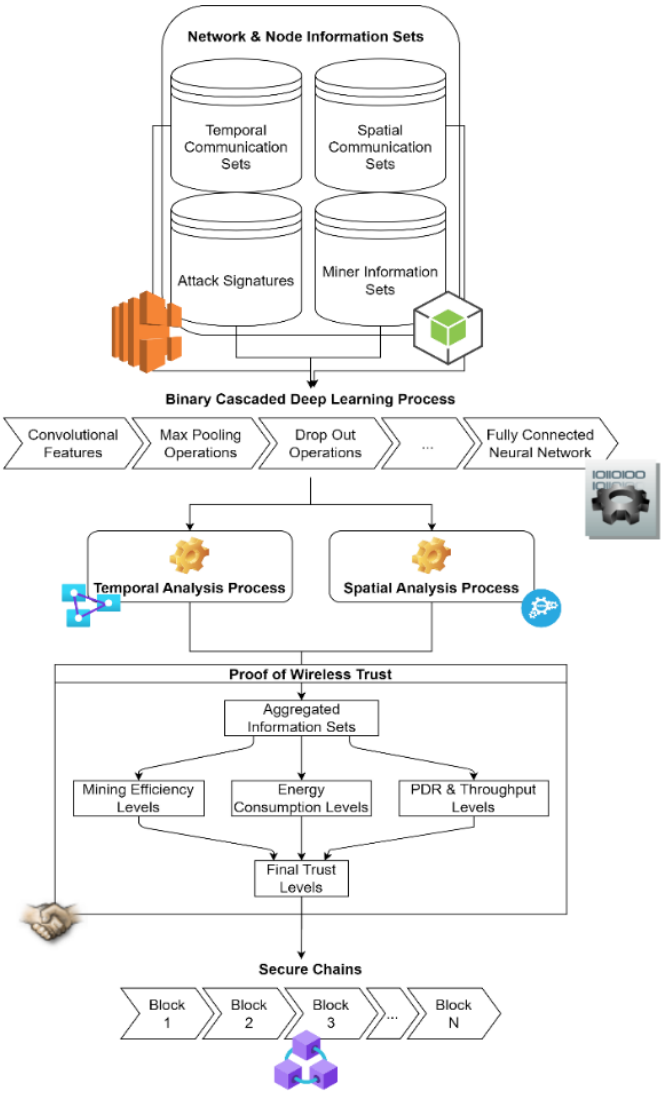

Figure 1 Design of the proposed Proof of Wireless Trust (PoWT)-based secure blockchain with attack detection capabilities

These scenarios include 'attack', and 'normal' communications. This distance metric is evaluated via equation 1, where, P represents packet signature which is estimated via equation 2, while s represents the network scenario under which this distance metric is evaluated for different packet types.

$$d(sa, sn) = \sqrt{\frac{\sum_{i=1}^{N_s}(P(sa)_i - P(sn)_i)^2}{N_s * Var(s_i, s_j)}} \dots (1)$$

Where, sn & sa represents normal & attack scenarios, while var represents variance between the signals, and is estimated via equation 3, and N_s represents total number of sample communications for which these evaluations are performed under different attack types. The variance is estimated by calculating mean (represented by $\bar{x}$) of input patterns.

$$P(x) = \frac{E(x) * D(x)}{THR(x) * PDR(x)} \dots (2)$$

Where, E & D represents the energy & delay needed to perform communications, while THR & PDR represents the throughput and Packet Delivery Ratio obtained during these communications.

$$var(x, y) = \sqrt{\frac{\sum_{i=1}^{N_s} \left(P(x) - \overline{P(x)}\right)^2 + \left(P(y) - \overline{P(y)}\right)^2}{2 * N_s}} \dots (3)$$

Based on this evaluation, a distance threshold is estimated via equation 4,

$$d_{th} = \sum_{i=1}^{NN}\sum_{j=1}^{NA} \frac{d(sn_i, sa_j)}{NN * NA} \dots (4)$$

Signatures with $d(sa, sa) > d_{th}$ are grouped into 'attack' category, while others are grouped into 'non-attack' category, and are processed via a Convolutional Neural Network (CNN) for identification of different attacks. The CNN Model is depicted in figure 2, and initially calculates convolutional features from packet signatures via equation 5,

$$Conv(P) = \sum_{a=0}^{\frac{m}{2}} P(i-a) * ReLU\left(\frac{m}{2} + a\right) \dots (5)$$

Where, m, a represents different window sizes, and stride sizes, while $ReLU$ (or Rectilinear Unit) is an activation function which is used to process negative feature sets via equation 6,

$$ReLU(x) = Max(0, x) \dots (6)$$

where x is the input to the rectilinear unit, and $f(x)$ is the set of outputs. The ReLU function returns x if x is greater than or equal to 0, and returns 0 if x is less than 0 which assists in removing negative feature sets. The ReLU function is used in CNNs because it is simple to compute and does not suffer from the vanishing gradients that can occur with other activation functions like the sigmoid functions. It has also been shown to improve the performance of deep neural networks on a variety of tasks.

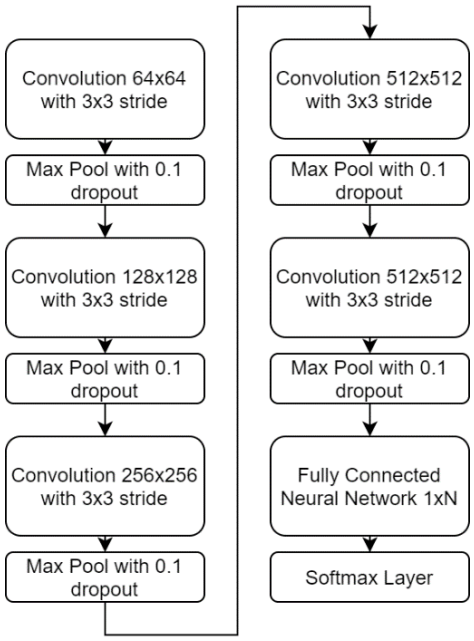

Figure. 2 Design of the customized CNN Model for identification of attacks

The window size & stride size is varied between 1x16, 1x32, 1x64, 1x128, 1x256, and 1x512 due to which the model evaluates high-density convolutional features. Total count of features can be estimated via equation 7 as follows,

$$f_{out} = \frac{f_{in} + 2 * p - k}{s} + 1 \dots (7)$$

Where, f_{in}, f_{out} represents input features & output features from different convolutional layers, while p, k & s represents their respective padding, kernel & stride sizes. Due to extraction of a large number of features, inherent redundancies are observed at output, which are removed by Max Pooling operations. The Max Pooling layer estimates a feature threshold via equation 8 as follows,

$$f_{th} = \left(\frac{1}{N} * \sum_{x \in N} x^p\right)^{\frac{1}{p}} \dots (8)$$

Where, N are total number of features, while p is the probability variance of current feature, which is estimated via equation 9,

$$p = \frac{x(p) - \overline{x(p)}}{Max(f)} \dots (9)$$

After evaluation of threshold, if the extracted feature value $f > f_{th}$, then the feature is retained and passed to next layers, else it is discarded due to low variance levels. This process is repeated for multiple convolutional layers and a large set of features are extracted during this process. At the final layer, a Fully Connected

Neural Network (FCNN) is used, which uses SoftMax for classification of features into binary attack classes (c_{out}) via equation 10,

$$c_{out} = \text{SoftMax}\left(\sum_{i=1}^{N_f} f_i * w_i + b_i\right) \dots (10)$$

Where, N_f are the total aggregated features evaluated by the convolutional layers, and w, b are tuneable weights & tuneable biases for these features. For N attack classes, $N-1$ such CNNs are used, which assist in finding final attack types. Individual CNNs are trained for 'non-attack', and 'single attack' categories, and final classification is done via equation 11,

$$c_{final} = \text{Non Attack, if converge}$$
$$\text{else,} \bigvee_{i=1}^{N_a} C_i \dots (11)$$

Where, N_a are total number of attacks for which the model is trained, C_i represents the output attack class obtained via CNN process, while $\bigvee C$ are intersection of different attack classes. Once these attack classes are obtained for individual nodes, then nodes which are under attack are removed from the communication process. For all other nodes, different attacks are simulated for a dummy set of communications. For each of these attacks, the communication delays are estimated via equations 12 & 13 as follows,

$$D(N) = \frac{\sum_{i=1}^{N(r)} t_{complete_i} - t_{start_i}}{N(r)} \dots (12)$$

$$D(A) = \frac{\sum_{i=1}^{A(r)} t_{complete_i} - t_{start_i}}{A(r)} \dots (13)$$

Where, $t_{complete}$ & t_{start} represents timestamps for completion & start of different communications, while $N(r)$ & $A(r)$ represents normal & attack communication requests. Similarly, the energy needed during these communications is estimated via equations 14 & 15 as follows,

$$E(N) = \frac{\sum_{i=1}^{N(r)} E_{start_i} - E_{complete_i}}{N(r)} \dots (14)$$

$$E(A) = \frac{\sum_{i=1}^{A(r)} E_{start_i} - E_{complete_i}}{A(r)} \dots (15)$$

Where, E_{start} & $E_{complete}$ represents residual energy of nodes during start and completion of different communications. This performance is further augmented via estimation of throughput & PDR levels during attack & normal scenarios. This is done via equations 16, 17, 18 & 19 as follows,

$$T(N) = \sum_{i=1}^{N(r)} \frac{Rx(P)_i}{N(r) * D(N)} \dots (16)$$

$$T(A) = \sum_{i=1}^{A(r)} \frac{Rx(P)_i}{A(r) * D(A)} \dots (17)$$

Where, $Rx(P)$ represents total number of packets received during different communications.

$$PDR(N) = \sum_{i=1}^{N(r)} \frac{Rx(P)_i}{N(r) * Tx(P)_i} \dots (18)$$

$$PDR(A) = \sum_{i=1}^{A(r)} \frac{Rx(P)_i}{Tx(P)_i * A(r)} \dots (19)$$

Where, $Tx(P)$ represents total number of packets transmitted during different communications. As per these evaluations, Proof-of-Wireless Trust (PoWT) is calculated via equation 20 as follows,

$$PoWT = \frac{\frac{E(N)}{E(A)} + \frac{D(N)}{D(A)} + \frac{PDR(A)}{PDR(N)} + \frac{T(A)}{T(N)}}{4} \dots (20)$$

This metric is evaluated for each of the 'non-attack' nodes, and based on it a trust threshold is calculated via equation 21,

$$T_{th} = \sum_{i=1}^{N(NA)} \frac{PoWT_i}{N(NA)} \dots (21)$$

Miner nodes with $PoWT > T_{th}$ are used for mining, while others are discarded from the mining process. Due to this selection, the model is able to identify high trust nodes, and use them for different mining operations. This selection assists in identification of attacks, and improves Quality of Service (QoS) levels for large-scale networks. Performance of this model is evaluated in the next section of this text, where it is compared in terms of different security & QoS levels with existing security models under different attacks.

RESULT EVALUATION AND STATISTICAL COMPARISONS

The proposed model employs a novel Proof-of-Wireless-Trust (PoWT) consensus algorithm to validate and secure the training data, as well as a customized Binary Cascaded Deep Learning Model (BCDLM) to train the model with respect to multiple attack signatures. The blockchain-based model is intended to detect and mitigate real-time dynamic network attacks, thereby improving the security of network systems. In terms of accuracy, computational efficiency, and security, the proposed model outperforms existing deep learning models. This is done via estimation of attack detection Accuracy (A), precision (P), recall (R), and delay (D) needed to evaluate different attacks. These metrics are calculated via equations 22, 23, 24 & 25 as follows,

$$A = \frac{1}{NC} \sum_{i=1}^{NC} \frac{t_{p_i} + t_{n_i}}{t_{p_i} + t_{n_i} + f_{p_i} + f_{n_i}} \dots (22)$$

$$P = \frac{1}{NC}\sum_{i=1}^{NC}\frac{t_{p_i}}{t_{p_i}+f_{p_i}} \dots (23)$$

$$R = \frac{1}{NC}\sum_{i=1}^{NC}\frac{t_{p_i}}{t_{p_i}+t_{n_i}+f_{p_i}+f_{n_i}} \dots (24)$$

$$d = \frac{1}{NC}\sum_{i=1}^{NC} ts_{complete_i} - ts_{start_i} \dots (25)$$

Where, t & f represents true and false rates. This performance is evaluated for NC communications under the following data samples,

- UNSW-NB15 Dataset Samples (https://zenodo.org/record/4519767)
- SNMP 2016 Dataset Samples (https://data.mendeley.com/datasets/krbhsg5xrt)
- Network Intrusion Prevention System Dataset Samples (https://www.technavio.com)
- LU Flow Network Intrusion Detection Dataset Samples (https://www.kaggle.com)
- UNR-IDD Intrusion Detection Dataset Samples (https://www.kaggle.com)
- Gure KDDCup Dataset Samples (https://figshare.com)

These categories were merged so that DDoS, Finney, Masquerading, Sybil, Spoofing, and Spying attacks could be identified. Eight distinct assault teams were responsible for acquiring 1.5 million samples (including normal class). From this set, 10% of the accumulation was allocated to activities involving validation and evaluation, while 80% was allocated to activities involving training scenarios. This technique was used to evaluate the classification accuracy, and the results were contrasted in Table 1 with BI[6], GNN[23], and CNN[30] in terms of the data's dependability levels. In these evaluations, Number of Test Samples (NTS) were used as a base point for comparative analysis.

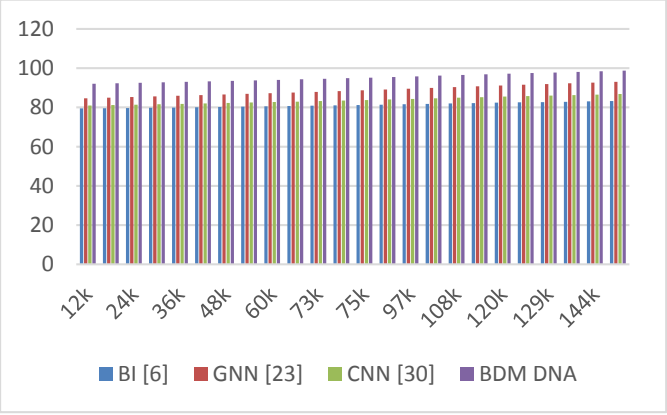

Figure. 3 Accuracy of attack classification under all 8 classes for different techniques

This assessment and figure 3 show that the suggested model can increase categorization accuracy by 15.5% when compared to BI,[6] 5.4% when compared to GNN,[23] and 12.4% when compared to CNN[30], making it helpful for a variety of real-time network situations. The incorporation of binary cascaded categorization using packet fingerprints has increased accuracy levels.

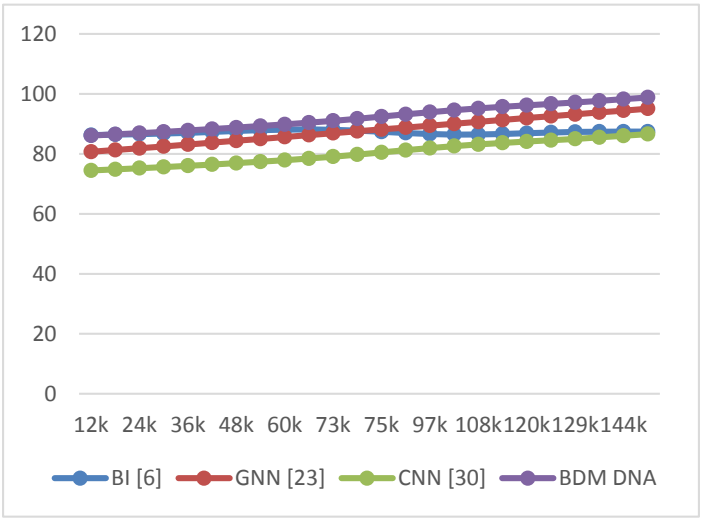

Figure. 4 Precision of attack classification under all 8 classes for different techniques

Figure 4 and the assessment show that compared to BI[6], GNN[23], and CNN[30], the suggested model improves categorization precision by 12.5%, 3.4%, and 10.5%, respectively, making it applicable to a broad range of real-time network situations. Packet signature analysis and binary categorization processes are combined to increase these precision levels.

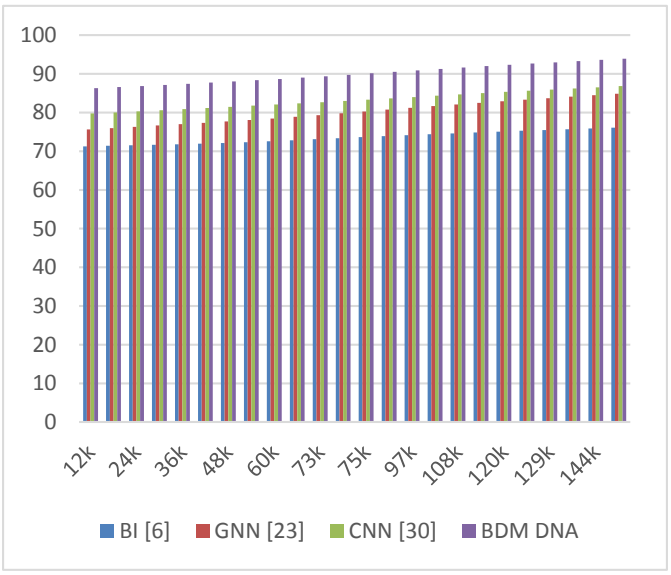

Figure. 5 Recall of attack classification under all 8 classes for different techniques

According to this assessment and figure 5, it can be seen that the suggested model is capable of improving the categorization recall by 15.5% when compared with BI,[6] 8.3% when compared with GNN,[23] and 6.5% when compared with CNN.[30] This enables it to

be helpful for a broad assortment of different real-time network situations. The combination of PoWT and binary categorization procedures has resulted in an improvement to recall levels.

Based on this evaluation and Figure 6, it can be seen that the proposed model can increase classification speed by 10.5% when compared to BI,[6] 18.3% when compared to GNN,[23] and 6.5% when compared to CNN,[30] making it applicable to a broad range of high-speed network scenarios. Integration of binary classification and packet signature analysis increases this rate. As a result of these enhancements, the proposed model is able to improve attack identification efficacy in both static and dynamic network scenarios.

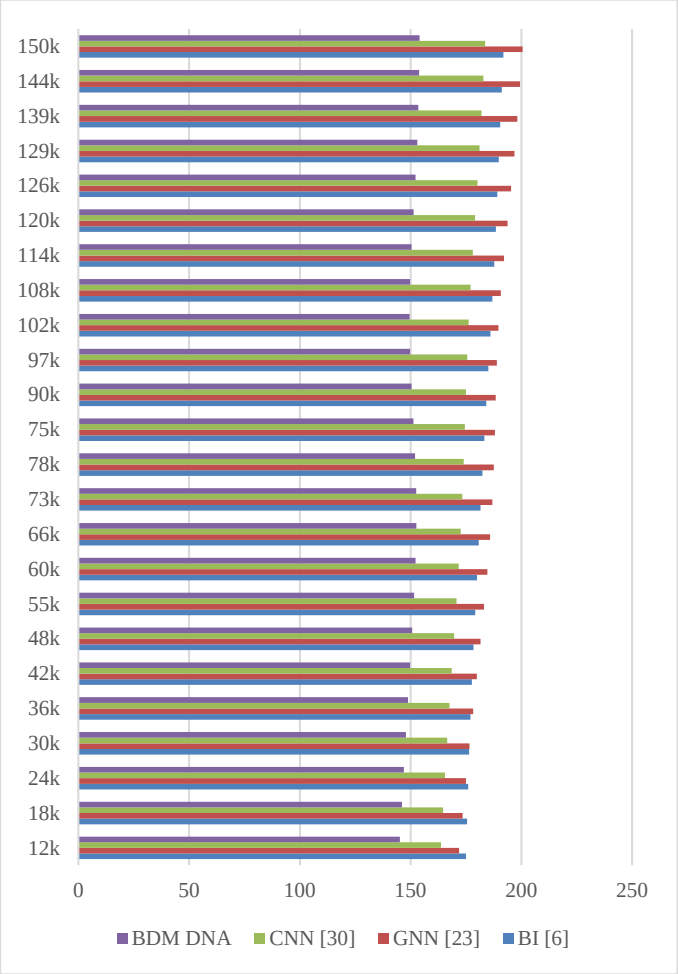

Figure. 6 Delay needed for attack classification under all 8 classes for different techniques

The proposed model's QoS performance is enhanced due to the incorporation of a PoWT blockchain model for data transmissions when compared to SVD,[31] LSTM CGAN,[32] and FL[39] models under various attack conditions. Adjusting the number of aggressor (NA) nodes from 1% to 15% and calculating the QoS values under Finney, Sybil, Masquerading, and DDoS attacks yields this performance. By combining PoWT and Binary Cascaded Deep Learning (BCDL), the model is capable of mitigating attacks while preserving higher QoS levels. The model was evaluated for 200k communications, and attack nodes were manipulated to evaluate various QoS metrics.

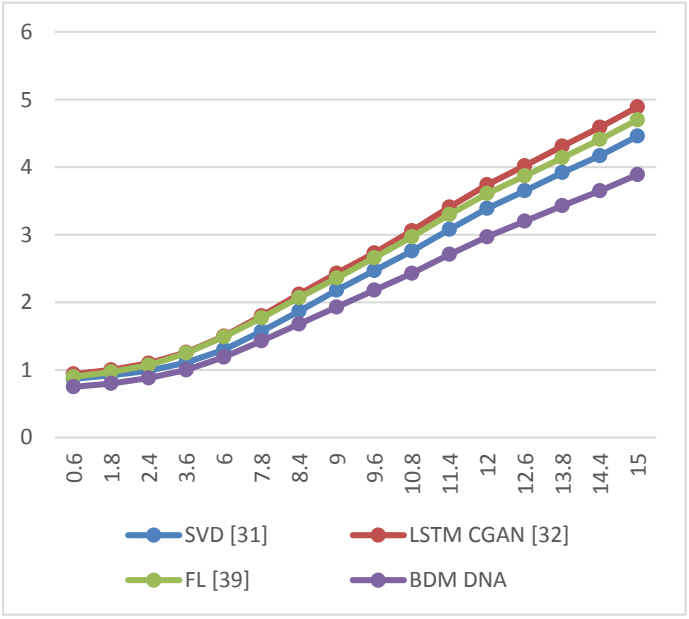

Figure. 7 Average communication delay for different attack scenarios

As per this evaluation and figure 7, it can be observed that the proposed model is able to improve the delay needed during communications by 10.5% when compared with SVD[31], 14.5% when compared with LSTM CGAZ[32], and 12.4% when compared with FL[39] under different attack levels. This performance is improved due to selection of high efficiency miner nodes via PoWT consensus that were evaluated for different attacks. Similarly, the complexity of mining (CM) (or energy needed for mining operations) was estimated.

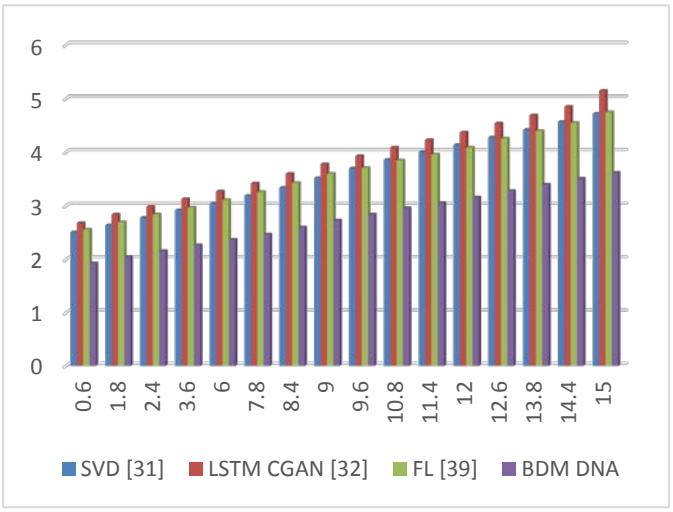

Figure. 8 Average complexity during mining for different attack scenarios

As per this evaluation and figure 8, it can be observed that the proposed model is able to improve the lifetime of network during

attack communications by 15.5% when compared with SVD[31], 18.3% when compared with LSTM CGAN[32], and 15.4% when compared with FL[39] under different attack levels. This energy performance is improved due to selection of high energy miner nodes via PoWT consensus that were evaluated for different attacks.

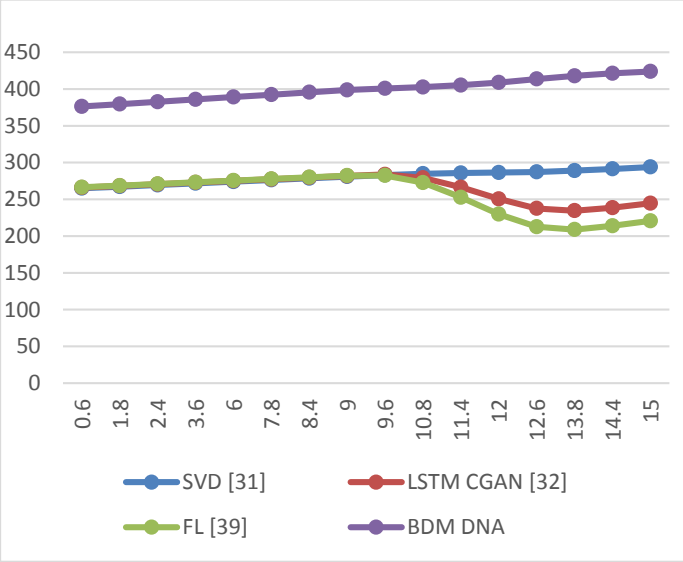

Figure. 9 Average throughput during mining for different attack scenarios

As per this evaluation and figure 9, it can be observed that the proposed model is able to improve the throughput of network during attack communications by 19.5% when compared with SVD[31], 24.5% when compared with LSTM CGAN[32], and 28.3% when compared with FL[39] under different attack levels. This throughput performance is improved due to selection of high throughput miner nodes via PoWT consensus.

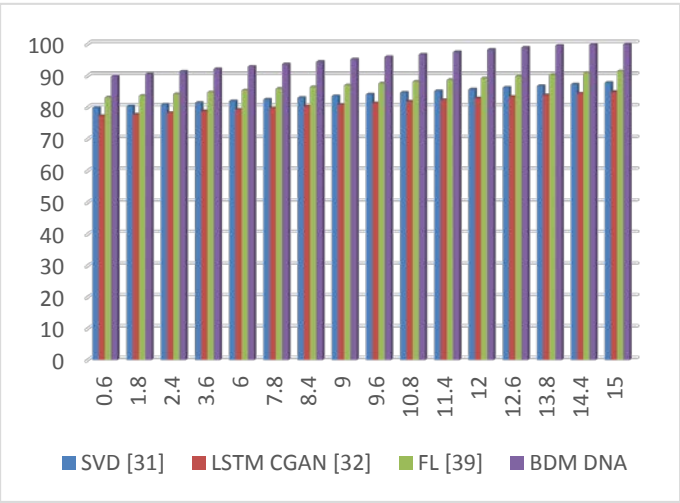

Figure. 10 Average PDR during mining for different attack scenarios

As per this evaluation and figure 10, it can be observed that the proposed model is able to improve the PDR of network during attack communications by 12.4% when compared with SVD[31], 15.5% when compared with LSTM CGAN,[32] and 6.5% when compared with FL[39] under different attack levels. This throughput performance is improved due to selection of high throughput miner nodes via PoWT consensus. Due to these optimizations, the proposed model is able to improve overall QoS of the network even under multiple attack types.

CONCLUSION

The proposed model employs a novel Proof-of-Wireless-Trust (PoWT) consensus algorithm and a customized Binary Cascaded Deep Learning Model (BCDLM) to train the model with respect to multiple attack signatures. The proposed model outperforms existing deep learning models in terms of accuracy, computational efficiency, and security. The incorporation of binary cascaded categorization and figure 3 demonstrate that the proposed model can improve categorization accuracy by 15.5% when compared to BI[6], 5.4% when compared to GNN,[23] and 12.0% when compared to CNN.[30] Figure 4 and the evaluation demonstrate that compared to BI,[6] GNN,[23] and CNN,[30] the proposed model improves categorization precision by 12.5%, 3.4%, and 10.0%. Combining packet signature analysis and binary categorization processes increases these levels of precision. The proposed model is capable of increasing the categorization recall by 15.5% when compared to BI[6], 8.3% when compared to GNN,[23] and 6.5% when compared to CNN[30] based on an evaluation of attack detection accuracy. This recall has been enhanced through the combination of PoWT and binary categorization procedures. Based on precision evaluation, the proposed model can increase classification speed by 10.5% compared to BI,[6] 18.3% compared to GNN,[23] and 6.5% compared to CNN.[30] Increasing this rate is the integration of binary classification and packet signature analysis. As a result of these enhancements, the proposed model is able to increase the efficiency of attack identification in both static and dynamic network scenarios.

Similarly, recall evaluation demonstrates that the proposed model can reduce the delay required for communications by 10.5% when compared to SVD,[31] 14.5% when compared to LSTM CGAN,[32] and 12.4% when compared to FL[39] at various attack levels. This performance is enhanced as a result of the evaluation and selection of high-efficiency miner nodes via PoWT consensus. In terms of evaluation delay, it can be seen that the proposed model improves network lifetime during attack communications by 15.5% when compared to SVD,[31] 18.3% when compared to LSTM CGAN,[32] and 15.4% when compared to FL[39] at various attack levels. This energy performance is enhanced as a result of the evaluation and selection of high-energy miner nodes via PoWT consensus that are resistant to a variety of attacks. In terms of network data rate, it can be seen that the proposed model improves the throughput of network communications during an attack by 19.5% when compared to SVD,[31] 24.5% when compared to LSTM CGAN,[32] and 28.3% when compared to FL[39] at various attack levels. This throughput performance is enhanced as a result of the selection of high throughput miner nodes using PoWT consensus.

The proposed model improves the PDR of network communications during an attack by 12.4% when compared to

SVD,[31] 15.5% when compared to LSTM CGAN[32], and 6.4% when compared to FL[39] at various attack levels. This throughput performance is enhanced as a result of the selection of high throughput miner nodes using PoWT consensus. Due to these optimizations, the proposed model is capable of enhancing the network's overall QoS even in the face of multiple attack types.

CONFLICT OF INTEREST STATEMENT

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

ACKNOWLEDGEMENT

I (NB) am grateful to all of those with whom I have had the pleasure to work during this research. Each of the members of my Dissertation Committee has provided me extensive personal and professional guidance and taught me a great deal about both research and life in general.

REFERENCES

1. V.N. Sathi, C.S.R. Murthy. Distributed Slice Mobility Attack: A Novel Targeted Attack Against Network Slices of 5G Networks. *IEEE Netw. Lett.* **2020**, 3 (1), 5–9.
2. Z. Zhai, G. Lai, B. Cheng, et al. Lightweight Secure Detection Service for Malicious Attacks in WSN With Timestamp-Based MAC. *IEEE Trans. Netw. Serv. Manag.* **2022**, 19 (4), 5299–5311.
3. S. Park, S. Kwon, Y. Park, D. Kim, I. You. Session Management for Security Systems in 5G Standalone Network. *IEEE Access* **2022**, 10, 73421–73436.
4. F.M. Atan, N. Zulkifli, S.M. Idrus, et al. Security enhanced dynamic bandwidth allocation algorithm against degradation attacks in next generation passive optical networks. *J. Opt. Commun. Netw.* **2021**, 13 (12), 301–311.
5. R. Harada, N. Shibata, S. Kaneko, et al. Quick Suppression of DDoS Attacks by Frame Priority Control in IoT Backhaul With Construction of Mirai-Based Attacks. *IEEE Access* **2022**, 10, 22392–22399.
6. N. Nishanth, A. Mujeeb. Modeling and Detection of Flooding-Based Denial-of-Service Attack in Wireless Ad Hoc Network Using Bayesian Inference. *IEEE Syst. J.* **2021**, 15 (1), 17–26.
7. Z.H. Pang, L.Z. Fan, Z. Dong, Q.L. Han, G.P. Liu. False Data Injection Attacks Against Partial Sensor Measurements of Networked Control Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, 69 (1), 149–153.
8. M. Pioro, M. Mycek, A. Tomaszewski. Network Protection against Node Attacks Based on Probabilistic Availability Measures. *IEEE Trans. Netw. Serv. Manag.* **2021**, 18 (3), 2742–2763.
9. O. Stan, R. Bitton, M. Ezrets, et al. Extending Attack Graphs to Represent Cyber-Attacks in Communication Protocols and Modern IT Networks. *IEEE Trans. Dependable Secur. Comput.* **2022**, 19 (3), 1936–1954.
10. Y. Li, Y. Zhao, J. Li, et al. DDoS Attack Mitigation Based on Traffic Scheduling in Edge Computing- Enabled TWDM-PON. *IEEE Access* **2021**, 9, 166566–166578.
11. N. Nishanth, A. Mujeeb. Modeling and Detection of Flooding-Based Denial of Service Attacks in Wireless Ad Hoc Networks Using Uncertain Reasoning. *IEEE Trans. Cogn. Commun. Netw.* **2021**, 7 (3), 893–904.
12. M. Zhang, L. Wang, S. Jajodia, A. Singhal. Network Attack Surface: Lifting the Concept of Attack Surface to the Network Level for Evaluating Networks' Resilience against Zero-Day Attacks. *IEEE Trans. Dependable Secur. Comput.* **2021**, 18 (1), 310–324.
13. S. Xiao, X. Ge, Q.L. Han, Y. Zhang. Secure Distributed Adaptive Platooning Control of Automated Vehicles Over Vehicular Ad-Hoc Networks Under Denial-of-Service Attacks. *IEEE Trans. Cybern.* **2022**, 52 (11), 12003–12015.
14. H. Guo, J. Sun, Z.H. Pang. Stealthy FDI Attacks Against Networked Control Systems Using Two Filters with an Arbitrary Gain. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, 69 (7), 3219–3223.
15. D. An, F. Zhang, Q. Yang, C. Zhang. Data Integrity Attack in Dynamic State Estimation of Smart Grid: Attack Model and Countermeasures. *IEEE Trans. Autom. Sci. Eng.* **2022**, 19 (3), 1631–1644.
16. Y. Li, S. Liu, Z. Yan, R.H. Deng. Secure 5G Positioning With Truth Discovery, Attack Detection, and Tracing. *IEEE Internet Things J.* **2022**, 9 (22), 22220–22229.
17. L. Yang, C. Wen. Optimal Jamming Attack System against Remote State Estimation in Wireless Network Control Systems. *IEEE Access* **2021**, 9, 51679–51688.
18. Q.Y. Fu, H.M. Wang. Detection of Hijacking DDoS Attack Based on Air Interface Traffic. *IEEE Wirel. Commun. Lett.* **2021**, 10 (10), 2225–2229.
19. Z. Wu, S. Peng, L. Liu, M. Yue. Detection of Improved Collusive Interest Flooding Attacks Using BO-GBM Fusion Algorithm in NDN. *IEEE Trans. Netw. Sci. Eng.* **2023**, 10 (1), 239–252.
20. Z. Wu, S. Peng, L. Liu, M. Yue. Detection of Improved Collusive Interest Flooding Attacks Using BO-GBM Fusion Algorithm in NDN. *IEEE Trans. Netw. Sci. Eng.* **2023**, 10 (1), 239–252.
21. D. Liu, D. Ye. Cluster Synchronization of Complex Networks under Denial-of-Service Attacks with Distributed Adaptive Strategies. *IEEE Trans. Control Netw. Syst.* **2022**, 9 (1), 334–343.
22. Y. Gao, M. Xu. Defense Against Software-Defined Network Topology Poisoning Attacks. *Tsinghua Sci. Technol.* **2023**, 28 (1), 39–46.
23. Y. Deng, H. Jiang, P. Cai, et al. Resource Provisioning for Mitigating Edge DDoS Attacks in MEC-Enabled SDVN. *IEEE Internet Things J.* **2022**, 9 (23), 24264–24280.
24. J. Liu, F. Labeau. Detection of False Data Injection Attacks in Industrial Wireless Sensor Networks Exploiting Network Numerical Sparsity. In *IEEE Transactions on Signal and Information Processing over Networks*; **2021**; Vol. 7, pp 676–688.
25. M. Zhang, G. Li, L. Xu, et al. Control Plane Reflection Attacks and Defenses in Software-Defined Networks. *IEEE/ACM Trans. Netw.* **2021**, 29 (2), 623–636.
26. A. Aljuhani. Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments. *IEEE Access* **2021**, 9, 42236–42264.
27. K. Abdelli, J.Y. Cho, F. Azendorf, et al. Machine-learning-based anomaly detection in optical fiber monitoring. *J. Opt. Commun. Netw.* **2022**, 14 (5), 365–375.
28. K. Abdelli, J.Y. Cho, F. Azendorf, et al. Machine-learning-based anomaly detection in optical fiber monitoring. *J. Opt. Commun. Netw.* **2022**, 14 (5), 365–375.
29. B.F. Yue, M.Y. Su, X.Z. Jin, W.W. Che. Event-Triggered MFAC of Nonlinear NCSs Against Sensor Faults and DoS Attacks. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, 69 (11), 4409–4413.
30. B. Ibrahim Hairab, H.K. Aslan, M.S. Elsayed, A.D. Jurcut, M.A. Azer. Anomaly Detection of Zero-Day Attacks Based on CNN and Regularization Techniques. *Electron.* **2023**, 12 (3), 98427–98440.
31. J. Xu, X. Li, P. Wang, X. Jin, S. Yao. Multi-Modal Noise-Robust DDoS Attack Detection Architecture in Large-Scale Networks Based on Tensor SVD. *IEEE Trans. Netw. Sci. Eng.* **2023**, 10 (1), 152–165.
32. Z. Liu, X. Yin. LSTM-CGAN: Towards Generating Low-Rate DDoS Adversarial Samples for Blockchain-Based Wireless Network Detection Models. *IEEE Access* **2021**, 9, 22616–22625.
33. A. Vangala, A.K. Das, A.K. Das, S.K. Das, Y. Park. Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, 18, 904–919.
34. S. Benadla, O.R. Merad-Boudia, S.M. Senouci, M. Lehsaini. Detecting Sybil Attacks in Vehicular Fog Networks Using RSSI and Blockchain. *IEEE Trans. Netw. Serv. Manag.* **2022**, 19 (4), 3919–3935.

35. P. Krishnan, K. Jain, K. Achuthan, R. Buyya. Software-Defined Security-by-Contract for Blockchain-Enabled MUD-Aware Industrial IoT Edge Networks. *IEEE Trans. Ind. Informatics* **2022**, 18 (10), 7068–7076.
36. D. Chattaraj, B. Bera, A.K. Das, J.J.P.C. Rodrigues, Y. Park. Designing Fine-Grained Access Control for Software-Defined Networks Using Private Blockchain. *IEEE Internet Things J.* **2022**, 9 (2), 1542–1559.
37. R. Yang, X. Chang, J. Misic, V. Misic, H. Kang. On Selfholding Attack Impact on Imperfect PoW Blockchain Networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, 8 (4), 3073–3086.
38. Z. Li, S. Gao, Z. Peng, et al. B-DNS: A Secure and Efficient DNS Based on the Blockchain Technology. *IEEE Trans. Netw. Sci. Eng.* **2021**, 8 (2), 1674–1686.
39. I. Aliyu, M.C. Feliciano, S. Van Engelenburg, D.O. Kim, C.G. Lim. A Blockchain-Based Federated Forest for SDN-Enabled In-Vehicle Network Intrusion Detection System. *IEEE Access* **2021**, 9, 102593–102608.