Graph Machine Learning based Cyber Attack Detection for Mobile Tactical Networks

Keerthiraj Nagaraj† *Dennis Agnew*† *Pavan K Mangipudi*† *Allen Starke*† *Zixiang Nie*§ *Janise McNair*†
† Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA
§ Department of Electrical, Electronics, and Communications Engineering, University of South Florida; Tampa, USA
{k.nagaraj, dennisagnew, pavan.mangipudi, allen1.starke}@ufl.edu, znie@usf.edu, mcnair@ece.ufl.edu

MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM) | 979-8-3503-2181-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/MILCOM58377.2023.10356310

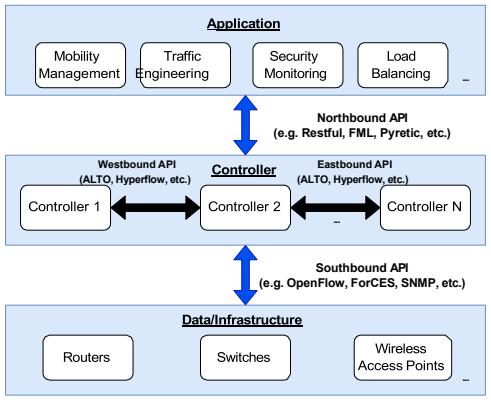
***Abstract*—First responders and other tactical teams rely on mobile tactical networks to coordinate and accomplish emergent time-critical tasks. The information exchanged through these networks is vulnerable to various strategic cyber network attacks. Detecting and mitigating them is a challenging problem due to the volatile and mobile nature of an ad hoc environment. This paper proposes MalCAD, a graph machine learning-based framework for detecting cyber attacks in mobile tactical software-defined networks. MalCAD operates based on observing connectivity features among various nodes obtained using graph theory, instead of collecting information at each node. The MalCAD framework is based on the XGBOOST classification algorithm and is evaluated for lost versus wasted connectivity and random versus targeted cyber attacks. Results show that, while the initial cyber attacks create a loss of 30%–60% throughput, MalCAD results in a gain of average throughput by 25%–50%, demonstrating successful attack mitigation.**

***Index Terms*—machine learning, cyber attacks, graph machine learning, intelligence, software-defined networking**

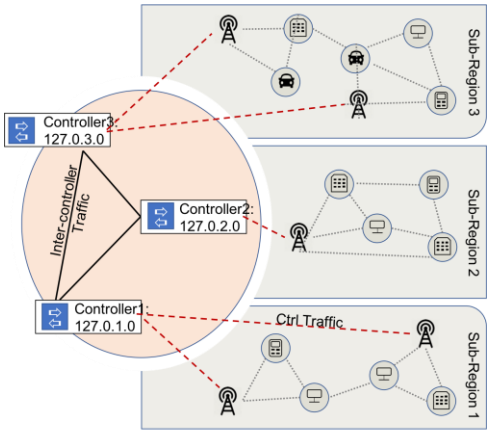
I. Introduction

Tactical mobile networks play an important role in providing communication capabilities for real-time situational awareness in emergency response scenarios for first responders. Emergency Medical Technicians (EMTs), paramedics, firefighters, police officers, remotely located soldiers, explorers, and other ad hoc communication teams, rely on tactical networks to provide a reliable communications architecture [1]. These networks are often made up of low-powered, battery-operated devices with ad hoc network connections, which bring several major challenges. First, tactical teams often move at a variety of speeds and directions, with continuous disruption and reestablishment of connections between the network's nodes or devices. Second, the condition of limited radio bandwidth and energy resources are amplified by the dynamically changing topology and the resulting changes in connectivity. Furthermore, due to their autonomous nature, ad-hoc networks are vulnerable to malicious cyber-attacks [2].

Characterization and detection of cyber threats in tactical mobile networks are crucial for secure real-time communication. Existing work on cyber attack detection relies on techniques that monitor performance data from network packets at each network node. These are computation and energy-intensive, reducing the network's lifetime. It is vital to have a cyber attack detection framework that has both a global view of the network and a low-energy approach. Network management functions in the mobile tactical network are provided by software-defined



(a)



(b)

Fig. 1: (a) Example SDN Controller Architecture [3] (b) Software-Defined Wireless Network

wireless networking (SDWN). As shown in Figure 1(a), the goal of software-defined networking (SDN) is to separate the data plane of the network from the control plane to allow for improved control, visibility, and security in network topology formation and routing [4]. The emergence of SDN has altered the wireless networking paradigm, creating distributed control in SDWNs as shown in Figure 1(b). Although an analysis of SDWN is out of the scope of this paper, we assume SDWN capability in our mobile tactical network to accomplish the network management tasks [5], [6].

In this work, we propose a graph learning and SDN-based cyber attack detection and mitigation framework called Machine-

Learning based Cyber Attack Detection (MalCAD). MalCAD leverages SDN's global visibility, and network topology information to extract connectivity features. Then, supervised machine learning models are trained to identify cyber threats, and SDN approaches are employed to restructure the network connections to circumvent compromised links and prevent data loss. Thus, this paper makes the following contributions:

- A new approach is proposed to integrate supervised machine learning and mobility modeling.
- An experimental analysis is provided that distinguishes targeted attacks from random cyber attacks.
- The proposed MalCAD framework provides a technique that is effective against jamming, black hole attacks, energy drain attacks, and denial of service attacks.
- The MalCAD framework detects the listed attacks with high accuracy ($> 95\%$) and can maintain system throughput in the presence of attacks.

This paper is organized as follows. Section II describes the related work. In Section III, we describe the MalCAD system framework and introduce the types of cyber attacks under consideration in this paper. In Section IV, we provide details of the MalCAD methodology, which includes the data collection process, the network connectivity features, and the application of the learning algorithm. Section V provides the simulation details, including the mobility model and dataset information, as well as the numerical results that show the impact of our framework on network performance. Finally, Section VI concludes the paper.

II. Related work

Using machine learning techniques for mobile tactical networks is still in its infancy stage. The published cyber attack detection frameworks focus on several categories, including highly-dynamic mobility [7], [8], and node behavior that can be attributed to successful cyber attacks, such as black holes, gray holes, and selfish greedy behaviors [9]. Authors of [10], [11], [12] attempt to reduce the spread of malware by reducing the contact between machines with similar vulnerabilities [10] or by monitoring large deviations in network performance statistics [11], [12]. The authors in [13] use packet arrival rates to detect intrusions using a support vector machine learning algorithm. This scheme does not apply to highly dynamic networks. Other works follow more unique paths in detecting and mitigating cyber attacks within the network. [14] provides a multi-stage anomaly detection method that uses external sources of information, beyond the conventional signatures and moni- tored network data, provides expert knowledge and contextual information, and demonstrates improved efficiency in intrusion detection. Authors of [15] propose a novel cross-stack sensor framework for attacker disinformation, misdirection, monitoring, and analysis. The network is protected by introducing "booby- traps" at network endpoints, operating systems, and application layers.

While the related work currently relies on monitoring and collecting data about each node as a key part of the dataset, the MalCAD approach does not need to do this intense monitoring of every node. Node-by-node monitoring can be difficult to obtain, may incur a large amount of processing, and may not give a clear picture of how the node is currently impacting the network. MalCAD monitors the connectivity characteristics, gains a more accurate picture of the impact, and thereby preserves energy, bandwidth, and privacy.

III. Cyber Attack Characterization

Mobile ad hoc networks are vulnerable to a range of cyber attacks [16], [17]. The attacks considered in this study and the expected impacts are as follows:

- **Jamming**: The attacker intentionally disrupts the reception of data to/from a node by interrupting the data transmission. *Impact*: Network traffic is reduced due to lost connections at the attacked node(s).
- **Blackhole:** The compromised, or corrupted, node receives packets that it discards instead of routing to the next node. *Impact*: Reduced network traffic, disrupted packet forwarding, network isolation, and changes in the topology of the network.
- **Energy drain attack**: The attacker sends connection establishment requests continuously to the attacked node. *Impact*: The incoming requests cause repeated receiving and processing of messages unnecessarily, resulting in depleted battery power and the loss of connections to the surrounding nodes. It also decreases network traffic and changes the width.
- **Denial of Service**: Attackers use excessive service requests to use up the resources of attacked nodes. *Impact*: Could significantly increase or decrease network traffic, and change the network topology. Distributed Denial of Service (DDoS) has a larger impact since it comes from different nodes.

We define two groups of cyber attacks based on the following criteria:

1) **Group 1 (Lost Connectivity)**: This group consists of attacks that **reduce** network traffic and cause attacked nodes to lose communication with their neighbors. Jamming and Blackhole are examples of this type of attack.
2) **Group 2 (Wasteful Connectivity)**: This group consists of attacks that **increase** network traffic and cause attacked nodes to communicate with a large number of nodes, resulting in a waste of network resources. Energy drain and Denial of Service are examples of this type of attack.

Group 1 (Lost Connectivity) and Group 2 (Wasteful Connectivity) attacks are both characterized by their impact on changes in connectivity among nodes and network topology. Hence, to detect these attacks, our methodology focuses on the features that capture the network connectivity and topology information. In the next section, we will describe the methodology of the proposed framework.

IV. MalCAD methodology

The proposed MalCAD framework is shown in Fig. 2. The system uses SDN-based network management to collect information on link status. This information is modeled using a graph

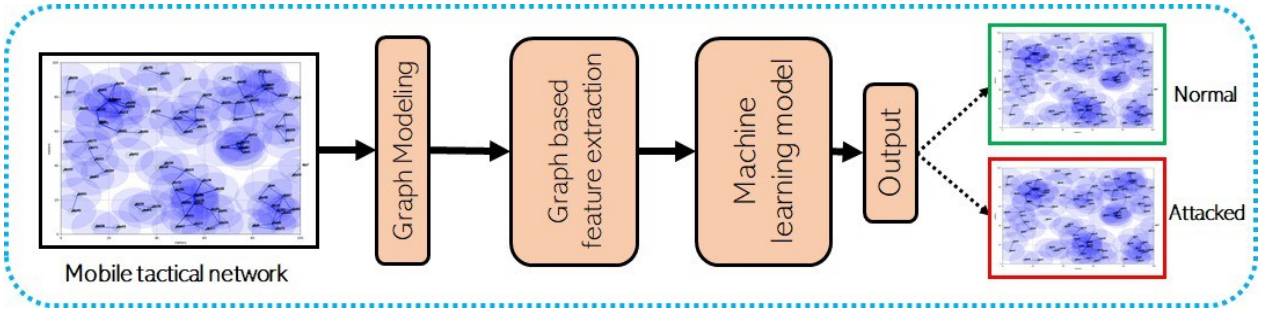


Fig. 2: MalCAD system diagram with individual components

to collect graph-based feature values (articulation points, clustering coefficients, bridge edges, and connected components). For the classifier, we use XGBoost to understand the variation of feature values that correspond to normal and attacked network behavior. XGBoost is a scalable end-to-end tree boosting system, which is used widely by data scientists [18]. Detection events will trigger the SDN network controller to update the network topology by isolating attacked nodes. We predict the topology change will increase network performance (restore the throughput) or reduce the waste of network resources. We also anticipate that this approach will be effective against jamming, black holes, energy drains, denial of service, and a variety of other cyber attacks.

The proposed MalCAD framework consists of three components: 1) data generation, 2) feature extraction, and 3) the machine learning algorithm used to develop the cyber attack detector models. As mentioned previously, the implementation of the mitigation steps will be conducted through the network control provided in the SDWN architecture.

A. Data generation

First, the position of the nodes in the tactical network is initialized. After initializing the node positions, the nodes are moved according to a selected mobility model that is representative of the behavior of nodes in a mobile tactical network. Then, we simulate the tactical network behavior for one day to extract feature values. Samples are collected, where each sample corresponds to a set of feature values extracted at a particular time instance.

The data collection process used in our analysis is shown in Procedure 1, where t refers to the sample number and T is the total number of samples. If there is an attack, we first determine if it is a Group 1 attack (Lost Connectivity) or a Group 2 attack (Wasted Connectivity). Next, we determine if the attack is targeted, i.e., a purposeful attack on the influential nodes in the network, or random, i.e., randomly selected nodes are attacked. Targeted and Random attacks are designed based on the analysis provided in [19]. The value *'m'* is the number of nodes that are attacked in the network for a given sample. Finally, the network response to the attack is to either remove the links of selected nodes to indicate a loss of connectivity or to increase the range of selected nodes to indicate an abrupt increase in network connectivity.

We collected datasets for different combinations of attack groups and attack types and generated cyber attack detection models using each combination.

Procedure 1 Data Collection

Initialize network simulation parameters.
Input: Initial node positions, Mobility model, Attack Type (**AT**), Attack Group (**AG**), Number of nodes to attack (m), Total samples in simulation (T).
for sample number $t = 1 : \mathbf{T}$ **do**
  Update node positions based on the mobility model
  if AG == Group 1 & AT == targeted **then**
    Remove links from m most influential nodes
  else if AG == Group 1 & AT == random **then**
    Remove links from m randomly selected nodes
  else if AG == Group 2 & AT == targeted **then**
    Increase range of m most influential nodes
  else if AG == Group 2 & AT == random **then**
    Increase range of m most randomly selected nodes
  else
    No changes to network topology
  Extract graph-based features described in Section IV.B for current network topology
Output: Graph-based feature datasets for attack detection

B. Feature Extraction: Network connectivity and graph-based topology features

Selecting the right machine learning features is key to implementing an effective detection strategy. To detect the various Group 1 (Lost Connectivity) and Group 2 (Wasted Connectivity) attacks, we require features that capture the network node connectivity and topology status. We employ graph-theoretic properties such as articulation points, bridge edges, strongly connected components, and clustering coefficients to model and measure the characteristics, calculating the graph theory metrics from each node's neighbor list. This reduces the need to monitor performance data at each node, which is computationally intensive.

- **Number of articulation points**: A given node is considered as an articulation point [20] if its removal splits the network into two or more disconnected components.

The higher the number of articulation points, the more vulnerable the network is to any targeted attacks.

- **Number of bridge edges**: A given edge is considered a bridge edge if its removal splits the network into two or more disconnected components. Similar to articulation points, the higher the number of bridge edges, the more vulnerable the network is to any targeted attacks.
- **Average Clustering Coefficient (ACC)**: The ACC is the ratio of closed triplets to all triplets (open and closed), where a triplet is a collection of three nodes. Any three nodes connected by two undirected ties form an open triplet, and any three nodes connected by three undirected ties form a closed triplet. A high ACC [21] indicates the network's ability to form strong and large communities within the network, helping to manage topology changes.
- **Strongly Connected Components**: A strongly connected component is a part of the network where every node can contact every other node through single or multiple hops. The size of the strongly connected components impacts the network performance to a great extent because of the large number of connections.

C. Machine Learning model

The final component of the MalCAD framework is a classification model, suited for classifying the state of the network, i.e., whether it was attacked or not. In our study, we chose a decision tree classifier [22] as a simple, yet efficient, machine learning technique. The decision tree classifier has hyper-parameters, such as the maximum length of the tree (depth), function to measure the quality of the split, and strategy to choose the split at each node (the options being to choose either the best split or random split). Extreme Gradient Boosting (XGBoost) [23], [18] with decision trees is an example boosting technique that uses decision trees as the base estimator. XGBoost has a fewer number of hyper-parameters than other data and computation-intense machine learning algorithms such as the Multi-Layer Perceptron Neural Networks (MLPNN). Boosting techniques give more importance to samples that have bad predictions and try to change model parameters to address them over several iterations. The iterations improve the overall model performance.

V. Simulation

A. Simulation Setup

A network simulation testbed was developed using Python in the Anaconda platform. We used Stanford Network Analysis Platform (*SNAP*) [24], a large-scale network analysis tool, and other popular Python library packages such as Pandas [25], Scikit-learn [26], and PyMobility [27] to carry out the analysis. We initialized the network with $N = 100$ nodes, with the nodes placed randomly in an area with dimensions 100 meters x 100 meters. Each node represents a device in the network. All the nodes have the same communication range, $R = 10$ meters, and only devices within range of each other are allowed to communicate. The simulator visualization of the network

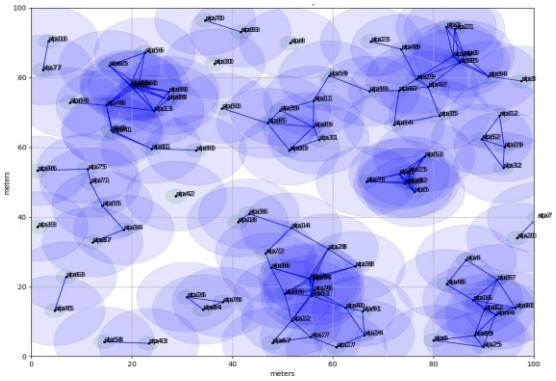


Fig. 3: A snapshot of the network from the simulation showing connections between different nodes and their range

topology with the aforementioned parameters is shown in Fig 3, which shows 100 nodes and transmission ranges.

Once the position and range of the nodes are initialized, the simulator sets up a mobility model for each node. This study considers the Reference Point Group Mobility (

06+ 8) model [28]. In the *RPGM* model, each collection of nodes has a logical center whose mobility follows a ran- dom way-point model. The non-logical-center nodes follow the logical-center node with some variance. The *RPGM* paradigm is a model that has been applied for emergency medical technician (EMT) teams, teams of soldiers on the battlefield, and groups in expeditions. As mentioned before, we begin with the initial node positions and then use the *RPGM* mobility model to move the nodes and extract feature values every 5 seconds. Each dataset has 17, 280 samples, where each sample contains feature values from a specific period. Each sample of the dataset corresponds to the feature values measured at a given time instance. We have two different kinds of time instances: 1) attacked time instances ('attacked') and 2) normal time instances ('normal'), which form the two classes for classification models. We will continue to use the phrases 'normal' for the cases when the network was not attacked and 'attacked' for the cases when the network was attacked, respectively. We first randomized the samples and then used 70% of the samples (12, 096) to train, and 30% of the samples (5, 184) to test the effectiveness of the attack detection models. A pre-processing phase normalized all the features to avoid any one factor unjustly influencing model training.

Once the dataset is generated and features are extracted, the XGBOOST model is used to classify the state of the mode. Using the parameter set mentioned in Section IV-C, we use Grid search K-fold cross-validation to select the set of optimal hyper-parameters from a range of possible values for each. In K-fold cross-validation, data is initially split into 'K' parts. In each step, 'K-1' parts are used for training the model and the remaining samples are used for testing the model performance. We then train models for all combinations of a set of hyper-parameter values and choose the model that performs the best during K- fold cross-validation. This K-fold process results in models that can generalize well with unseen data. In our experiments, we

found the optimal number of estimators parameter to be 50 and the maximum tree depth parameter to be 10. Other parameter values were left as their default values as set in XGBoost library [23], [18].

B. Numerical Results

Our proposed framework's cyber attack detection performance is evaluated utilizing Accuracy, Precision, Recall, and F1-score [29].

Tables I- IV show the performance results for the cyber attack detection models developed using the proposed MalCAD framework. Table I and Table II show the performance of MalCAD for Group 1 attacks (Lost Connectivity). Table I shows results for Random cyber attacks and Table II shows targeted Group 1 attacks. Table III and Table IV show the performance of the MalCAD framework for Group 2 (Wasted Connectivity). Table III shows results for Random cyber attacks and Table IV shows targeted Group 2 attacks.

TABLE I: Attack group - Group 1, Attack type - Random

attacked nodes (%)	Accuracy	Precision	Recall	F1-score
5	99.70	100.0	96.60	98.27
10	99.20	98.39	92.45	95.33
15	99.76	100.0	97.35	98.66
20	99.50	98.44	95.84	97.13

TABLE II: Attack group - Group 1, Attack type - Targeted

attacked nodes (%)	Accuracy	Precision	Recall	F1-score
5	99.93	100.0	99.24	99.62
10	100.0	100.0	100.0	100.0
15	99.93	100.0	99.24	99.62
20	99.80	98.50	99.24	98.87

TABLE III: Attack group - Group 2, Attack type - Random

attacked nodes (%)	Accuracy	Precision	Recall	F1-score
5	94.43	77.84	51.69	62.13
10	97.66	88.53	84.52	86.48
15	98.86	95.29	91.69	93.46
20	99.66	98.85	97.35	98.09

From Tables I-IV, we can observe that it is easier to detect targeted attacks in the network than random attacks, which is a favorable outcome as the targeted cyber attacks are more dangerous for network performance. Targeted attacks result in a better-performing model since the targeted nodes result in a higher impact on the node connectivity and topology, which can be captured more effectively using the features employed in the proposed framework. We can also observe that, as the number of attacked nodes increases, the attack detection model performs better due to the same reason.

TABLE IV: Attack group - Group 2, Attack type - Targeted

attacked nodes (%)	Accuracy	Precision	Recall	f1-score
5	94.23	72.33	56.22	63.26
10	97.86	90.03	85.28	87.59
15	98.56	95.12	88.30	91.58
20	99.60	98.84	96.60	97.70

C. Impact of cyber attack detection framework

In this section, we analyze the impact of MalCAD on network performance, using Mininet-WiFi, an open-source software-defined wireless network (SDWN) emulator. In this simulation, the controller (1) monitors the nodes and flows within the network; (2) extracts the features needed by MalCad to detect the attacks; (3) generates the mitigation routing solution, and (4) enforces the mitigation mechanism by refreshing the flow tables in the wireless access points or routers. The simulation environment consists of $N = 100$ nodes randomly placed in an area of 100×100 meters with each node having a range of $R = 10$ meters.

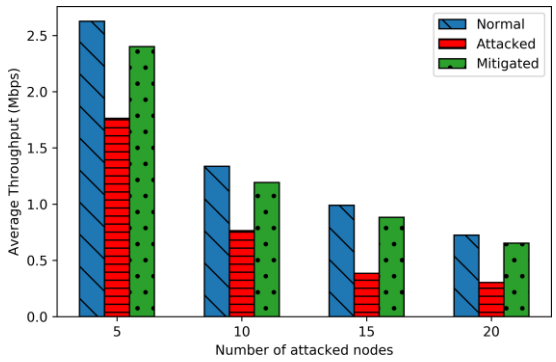


Fig. 4: Variation in average throughput for different numbers of attacked nodes

To perform attacks and gather the performance result, iPerf 3 (3.1.3) and hping3 (3.0.0-alpha-2) are used to benchmark and initiate attacks, respectively. Network throughput was monitored starting at the beginning of the simulation when the network only contains normal traffic. We then initiated a targeted cyber attack on m nodes in the network by sending a large number of TCP SYN packets to the attacked wireless and mobile nodes to drain their resources. We then examined the link behavior, extracted the features, and processed the results using the MalCAD model to see if it successfully detected and classified our TCP cyber attack. If MalCAD indicates an attack, a request is initiated at the SDWN controller to begin mitigation, which for this class of attack was to limit the flow of TCP SYN packets to the attacked nodes, thereby improving the network performance considerably.

In fig. 4, we show the variation of average throughput, i.e., the throughput of all the attacked nodes averaged over the entire simulation. The average throughput is shown for cases of 5, 10,

15, and 20 attacked nodes. For each case, we show the average throughput during normal traffic, attacked traffic, and mitigated traffic. When 5 nodes were attacked, the average throughput of the network initially dropped by as much as 32%, but after the MalCAD-initiated response, the throughput was restored and increased by 24%. For the case when 10 nodes were attacked, the average throughput initially decreased by 42% but MalCAD increased throughput by 32%. Similarly, for the cases when 15 and 20 were attacked, the initial average throughput decreased by 61% and 58% respectively, but due to the attack detection capability of MalCAD with the attack mitigation scheme, the average throughput increased by 50% and 48% respectively.

VI. Conclusions and Future Work

In this paper, we presented a graph machine learning-based cyber attack detection framework, called MalCAD. MalCAD exhibited the ability to identify cyber threats in mobile tactical networks. Our suggested strategy uses globalized network connectivity to avoid monitoring networking-related metrics at each node. MalCAD was evaluated for lost connectivity, wasted connectivity, and random, and targeted cyber attacks. Our platform detects targeted cyber attacks, which are more damaging to network performance. We showed that detecting a cyber attack prompts the network controller to act quickly, preventing additional network degradation. Due to MalCAD's attack detection capability, network throughput rose by 25-50%, compared to 30-60% caused by cyber attacks. In future work, we intend to incorporate additional realistic mobility models for tactical situations, novel types of cyber attacks, and the impact of other attack mitigation strategies into this framework.

Acknowledgment

This work was supported by the National Science Foundation under Grant Number 1738420.

References

[1] S. Teotia and S. Garg, "An effective and optimal mobility model and its prediction in manets," *International Journal of Trend in Scientific Research and Development*, vol. Volume-2, no. Issue-1, p. 1634–1642, 2017.

[2] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device," *Ad Hoc Networks*, vol. 84, pp. 82–89, 2019.

[3] D. Agnew, N. Aljohani, R. Mathieu, S. Boamah, K. Nagaraj, J. McNair, and A. Bretas, "Implementation aspects of smart grids cyber-security cross-layered framework for critical infrastructure operation," *Applied Sciences*, vol. 12, no. 14, p. 6868, 2022.

[4] S. H. Haji, S. R. Zeebaree, R. H. Saeed, S. Y. Ameen, H. M. Shukur, N. Omar, M. A. Sadeeq, Z. S. Ageed, I. M. Ibrahim, and H. M. Yasin, "Comparison of software defined networking with traditional networking," *Asian Journal of Research in Computer Science*, vol. 9, no. 2, pp. 1–18, 2021.

[5] M. Fogli, C. Giannelli, and C. Stefanelli, "Software-defined networking in wireless ad hoc scenarios: Objectives and control architectures," *Journal of Network and Computer Applications*, p. 103387, 2022.

[6] M. von Rechenberg, P. H. Rettore, R. R. F. Lopes, and P. Sevenich, "Software-defined networking applied in tactical networks: Problems, solutions and open issues," in *2021 International Conference on Military Communication and Information Systems (ICMCIS)*. IEEE, 2021, pp. 1–8.

[7] S. Zwane, P. Tarwireyi, and M. Adigun, "Performance analysis of machine learning classifiers for intrusion detection," in *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. IEEE, 2018, pp. 1–5.

[8] W. Pawgasame, "A survey of deep learning for tactical wireless networks," in *2018 5th Asian Conference on Defense Technology (ACDT)*. IEEE, 2018, pp. 37–44.

[9] A. M. Shabut, K. Dahal, M. S. Kaiser, and M. A. Hossain, "Malicious insider threats in tactical manet: The performace analysis of dsr routing protocol," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 390–395.

[10] J. R. Morris-King and H. Cam, "Controlling proximity-malware infection in diverse tactical mobile networks using k-distance pruning," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 503–508.

[11] K. F. Yu and R. E. Harang, "Machine learning in malware traffic classifications," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 6–10.

[12] M. Dasari, "Real time detection of mac layer dos attacks in ieee 802.11 wireless networks," in *2017 14th IEEE annual consumer communications & networking conference (CCNC)*. IEEE, 2017, pp. 939–944.

[13] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," *IEEE Access*, vol. 7, pp. 42 450–42 471, 2019.

[14] F. J. Aparicio-Navarro, T. A. Chadza, K. G. Kyriakopoulos, I. Ghafir, S. Lambotharan, and B. AsSadhan, "Addressing multi-stage attacks using expert knowledge and contextual information," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2019, pp. 188–194.

[15] F. Araujo, T. Taylor, J. Zhang, and M. Stoecklin, "Cross-stack threat sensing for cyber security and resilience," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2018, pp. 18–21.

[16] F. Abdel-Fattah, K. A. Farhan, F. H. Al-Tarawneh, and F. AlTamimi, "Security challenges and attacks in dynamic mobile ad hoc networks manets," in *2019 IEEE jordan international joint conference on electrical engineering and information technology (JEEIT)*. IEEE, 2019, pp. 28–33.

[17] I. Tomic´ and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1910–1923, 2017.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[19] K. Nagaraj, S. S. Bhasale, J. McNair, and A. Helmy, "Vulnerability assessment and classification based on influence metrics in mobile social networks," in *Proceedings of the 17th ACM International Symposium on Mobility Management and Wireless Access*, 2019, pp. 9–16.

[20] S. Bisht, K. Shekhawat, N. Upasani, R. N. Jain, R. J. Tiwaskar, and C. Hebbar, "Transforming an adjacency graph into dimensioned floorplan layouts," in *Computer Graphics Forum*. Wiley Online Library, 2022.

[21] I. Ullah, M. Manzo, M. Shah, and M. G. Madden, "Graph convolutional networks: analysis, improvements and results," *Applied Intelligence*, vol. 52, no. 8, pp. 9033–9044, 2022.

[22] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[23] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[24] J. Leskovec and R. Sosicˇ, "Snap: A general-purpose network analysis and graph-mining library," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 1, pp. 1–20, 2016.

[25] J. Reback, W. McKinney, J. Van Den Bossche, T. Augspurger, P. Cloud, A. Klein, S. Hawkins, M. Roeschke, J. Tratner, C. She *et al.*, "pandas-dev/pandas: Pandas 1.0. 5," *Zenodo*, 2020.

[26] E. Bisong, "Introduction to scikit-learn," in *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019, pp. 215–229.

[27] A. Panisson, "panisson/pymobility." [Online]. Available: https://github.com/panisson/pymobility

[28] P. D. Dorge and S. L. Meshram, "Design and performance analysis of reference point group mobility model for mobile ad hoc network," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2018, pp. 51–56.

[29] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2020.