Investigation of Multi-stage Attack and Defense Simulation for Data Synthesis

Ömer Sen*†, Bozhidar Ivanov*, Martin Henze ‡§, Andreas Ulbig *†,
*IAEW, RWTH Aachen University, Aachen, Germany & †DE, Fraunhofer FIT Aachen, Germany
Email: {oemer.sen, andreas.ulbig}@fit.fraunhofer.de, {o.sen, a.ulbig}@iaew.rwth-aachen.de, bozhidar.ivanov@rwth-aachen.de
‡SPICe, RWTH Aachen University, Aachen, Germany & §CAD, Fraunhofer FKIE, Wachtberg, Germany
Email: henze@cs.rwth-aachen.de

***Abstract*—The power grid is a critical infrastructure that plays a vital role in modern society. Its availability is of utmost importance, as a loss can endanger human lives. However, with the increasing digitalization of the power grid, it also becomes vulnerable to new cyberattacks that can compromise its availability. To counter these threats, intrusion detection systems are developed and deployed to detect cyberattacks targeting the power grid. Among intrusion detection systems, anomaly detection models based on machine learning have shown potential in detecting unknown attack vectors. However, the scarcity of data for training these models remains a challenge due to confidentiality concerns. To overcome this challenge, this study proposes a model for generating synthetic data of multi-stage cyber attacks in the power grid, using attack trees to model the attacker's sequence of steps and a game-theoretic approach to incorporate the defender's actions. This model aims to create diverse attack data on which machine learning algorithms can be trained.**

***Index Terms*—Intrusion Detection, Smart Grid, Cyberattacks, Cyber Security, Game Theory**

I. Introduction

A. Motivation & Background

The availability of power grids has become increasingly important in recent years due to the growing need for stable energy supply. However, the increased usage of Information and Communication Technologies (ICT) in distribution grids has introduced new possibilities and dangers, particularly in terms of cybersecurity attacks [1]. The 2015 cyberattack on Ukrainian regional Distribution System operator companies serves as an example of the potential disruption caused by such attacks [2]. With the increasing size and complexity of power grids, as well as the growing dependence on digitalization and renewable energy sources, safeguarding cybersecurity is essential to ensure the reliable functioning of power systems [3]. Intrusion Detection Systems (IDSs) are actively developed to enhance the protection of critical infrastructure by observing and scrutinizing network or system behavior to identify potential cyber threats [4]. Detecting and preventing planned attacks on the power system require the implementation of effective measures, and Machine Learning (ML) algorithms are being developed for this purpose. However, the lack of data can limit the predictability power of these algorithms, necessitating the generation of synthetic attack data that captures the characteristics of real attacks.

B. Relevant Literature

Several approaches have been explored for generating synthetic cyberattack data for smart grids, including the creation of lab environments such as the CICIDS2017 dataset [5], the development of the ID2T framework for reproducible datasets [6], modification of existing datasets like UNSW-NB15 using denoising autoencoders and Wasserstein Generative Adversarial Networks (GANs) [7], and the generation of artificial attack data for smart grid security using frameworks like Melody [8]. Furthermore, studies such as [9] demonstrate the potential of data-driven deep reinforcement learning for proactive cyber defense, while [10] introduces a cross-layered framework for securing the power grid. Generating comprehensive and realistic datasets for effective ML model training is challenging due to various factors such as infrastructure implementation, scenario development, and ensuring data integrity and privacy. Publicly available datasets may be limited and sharing data publicly can increase the risk of cyberattacks, highlighting the need for more comprehensive datasets. This work utilizes game theory [11] to model the dynamics between an attacker and defender in a power grid intrusion scenario, employing key terms such as "starting capital", "funds", "betweenness centrality", and "path optimization" to analyze their capabilities and strategic decision-making.

C. Contributions & Organization

This paper aims to develop a model for generating synthetic data of cyberattacks for ML-based IDS in power grids, considering the dynamics between attackers and defenders. Our contributions are:

1) We develop a model for generating synthetic cyberattack data for ML-based IDS.
2) We propose a method that incorporates attack tree modeling and game theory mechanics to generate diverse attack data.
3) We evaluate different ML models on the generated data and analyze the impact of attacker-defender dynamics on detection quality and attack complexity.

The structure of this paper is as follows: Section III delves into the methodology used to generate synthetic attack data, followed by the evaluation of the generated data and the ML models used for IDS in Section IV. The results are discussed, and conclusions are drawn in Section V.

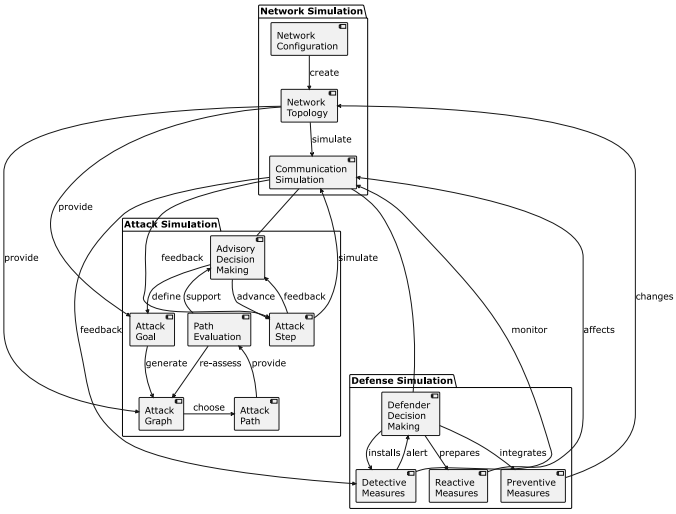

Fig. 1. Structural overview of the presented approach, which includes a game-theoretic simulation of the dynamic interplay of cyberattacks and defenses, with a focus on continuous learning and strategic adaptation.

II. Nomenclature

The mathematical symbols used in the equations throughout the paper are presented in Table I.

TABLE I
NOMENCLATURE

Symbol	Description
$W_{i,j}$	Weight of the edge connecting nodes i and j
t	Time needed for a particular step
$C_j^{attacker}$	Outage costs for the component from the attacker's viewpoint
$P_j^{attacker}$	Likelihood of successfully compromising a node
$Risk$	Risk of grid operation disruption due to cyberattack
Q_i	Learning rate for each node i
$c_{CB}(v)$	Current flow betweenness centrality
$\tau_{st}(v)$	Throughput from node s to node t via node v
$b_{st}(v)$	Absolute value of the total amount of current that flows through v
$r(\vec{e_{i,j}})$	Resistance between nodes i and j
c_i^{outage}, c_j^{outage}	Outage costs of nodes i and j
$TTC(s, W)$	Time to Compromise
t_1, t_2	Time taken for the first and second stages of an attack
P_1	Probability of the first stage of an attack
u	Unsuccessful rate of the second stage of an attack
N	Total number of nodes

III. Multi-Staged Attack & Defense Simulation

As our main contribution in this work, we present our method for generating synthetic cyberattack data using a game-theoretic approach between the attacker and defender.

A. Overview

Our work is based on a game-theoretic approach for modeling the dynamics of cyberattacks and defenses (cf. Figure 1). To understand how the dynamics between the attacker and defender affect data quality, we model a red vs. blue teaming approach in the simulation. The attacker aims to cause disruption to the grid operation, while the defender aims to prevent it. The attacker scans the system infrastructure and selects a target with high outage costs, while the defender takes preventative measures to lower the risk. We utilize procedural attack graph generation, which adapts at each game turn considering the defender's counter actions. The success rate is calculated based on prior knowledge, and the attacker chooses the shortest path using Dijkstra's Algorithm. Both the defender and attacker have learning rates that influence their strategies, and the attack path evolves over time. This learning effect is simulated by transferring knowledge from the previous simulation round to the next one. Critical network nodes were equipped with IDS, generating alerts for potential threats in a Python-based environment, where Multi-host, Multi-stage Vulnerability Analysis (MulVAL) execution was facilitated via Docker containers. The Python packages sklearn and xgboost were used to train ML models, with optimal hyperparameters determined via Grid Search to minimize overfitting.

B. Attack Graphs

Modeling an attack on a complex system like the power grid requires considering multi-stage attacks instead of just single compromises. Attack graph generation tools can model intricate situations involving simultaneous exploitation of multiple vulnerabilities, leading to a multi-stage cyberattack. These tools also take into account the environment, severity, and impact of the exploited vulnerabilities. Various attack graph generation tools were considered, with open-source tools providing more detailed attack paths but lacking user-friendliness due to poor visualization and obscurity. To ensure detailed attack graphs and scalability, open-source tools like MulVAL [12] were chosen for their scalability and extensibility. MulVAL is a logical attack graph generation tool that uses Datalog as an input language [12]. It utilizes existing vulnerability databases such as the National Vulnerability Database (NVD), allowing vulnerabilities to be specified by their ID. Host and network configurations can be provided by an Open Vulnerability and Assessment Language (OVAL) scanner and a firewall management tool, respectively [12].

C. Attacker and Defender

The attack and defense dynamics are modeled using a game-theoretic approach similar to [11]. The interaction between the attacker and defender is illustrated in Figure 2. We model the actions of the attacker and defender using the MITRE ATT&CK [13] and D3FEND [14] matrix. Initially, the attacker performs a network scan to identify the node with the highest outage costs. For instance, the Supervisory Control and Data Acquisition (SCADA) Server is highly susceptible to significant outage costs due to its expensive replacement and the potential for causing grid outages by compromising its security. The defender takes preventative and reactive measures to reduce the risk (cf. [14]). For example, the defender can use IDS as sensors or employ access restriction measures. IDS sensors are placed between devices to detect attacks. In the simulation, signature-based IDS sensors are used to represent the defender's detection capability, classifying non-conforming data as attack-generated based on predefined rules. These IDS sensors are solely used to represent the defender's

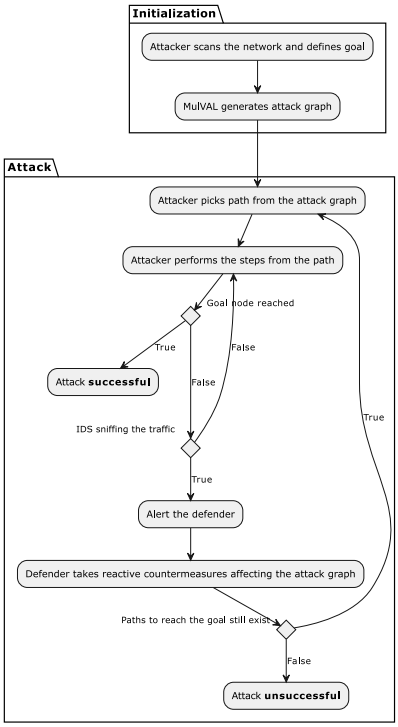

Fig. 2. Overview of the attacker-defense dynamic, showcasing the attacker scanning the network and the defender employing preventive and reactive measures, including signature-based IDS sensors for detection in the simulation.

detection scale in the simulation and are not considered in the later evaluation, where ML-based anomaly detection scores are used instead. The number of sensors is varied to observe its effect on the generated data. The MulVAL attack graph is created and transformed to represent the attacker's actions. The attacker follows a path, compromising components until either they reach their goal or an IDS detects their actions, triggering reactive countermeasures by the defender. This process continues until the attacker succeeds or the attack graph becomes non-traversable, resulting in an unsuccessful attack. Both the attacker and defender learn from previous experiences and take more optimal steps to improve the quality of attacks and overall strategy.

D. Attacker

The attacker aims to cause disruption to the grid operation while minimizing the risk of being detected by an IDS. The disruption caused to the grid operation is calculated based on the outage costs caused by the attacker's actions. To incrementally improve, the attacker increases their skill rate after each attack and updates their knowledge of the success rate of compromising a node. The skill rate determines the success of an attack, considering failures and action attempts, and is probabilistic in nature. Both factors influence the path chosen by the attacker, who uses Dijkstra's Algorithm to determine the optimal route [15]. The edge weights in the attack graph are computed using Equation 1, where $W_{i,j}$ represents the weight of the edge connecting nodes i and j:

$$W_{i,j} = \frac{t_j^{attacker}}{C_j^{attacker} \cdot P_j^{attacker}} \tag{1}$$

The superscript "attacker" indicates that the components of the equation are evaluated from the attacker's viewpoint. The variable t represents the time needed for a particular step, determined by the Time-to-Compromise (TTC) metric outlined in Section III-F. $P_j^{attacker}$ illustrates the likelihood of successfully compromising a node. Initially, the attacker is unaware of the actual success rate and assumes it to be 1. After each compromise attempt in both the current and previous attacks, a 1 or 0 is recorded to indicate success or failure, respectively. The mean of these values is then used for the calculation. If a random number between 0 and 1 is less than the actual success rate, the action can be executed by the attacker; otherwise, they must retry, spending more time and increasing the defender's resources, which allows for the implementation of additional preventative measures, further reducing the success rate. Alternatively, the attacker may reconsider the path if deterred by the updated success rate. The outage costs for the component from the attacker's perspective are denoted by $C_j^{attacker}$.

E. Defender

The defender's goal is to minimize the risk of grid operation disruption due to cyberattacks. They can implement preventative or reactive countermeasures, with preventative measures requiring time and funds to implement and operate, executed in subsequent simulation runs, while reactive measures can be executed instantaneously without capital, within the current run. The starting capital and funds for the defender represent the initial and acquired financial resources used to implement defensive measures against cyber threats. The defender enhances their knowledge by considering the attacker's most frequent pathways from previous attacks. This influences risk calculation and the corresponding preventative countermeasures. The calculation of the risk is defined in Equation 2, where the learning rate Q_i is added for each node i in the network topology graph, initially set to 1 and increasing with each attack if detected:

$$Risk = \sum_{i=1}^{N} P_i \cdot C_i \cdot Q_i \tag{2}$$

IDS sensor placement prioritizes nodes with higher outage costs. Centrality-based algorithms, such as current flow betweenness centrality, are used to identify important graph elements [16]. Betweenness centrality measures a node's contribution to information exchange, assuming information flows along the shortest path. Current flow betweenness centrality, which views the graph as an electrical network, removes this assumption:

$$c_{CB}(v) = \frac{1}{(n-1)\cdot(n-2)} \sum_{s,t \in V} \tau_{st}(v) \tag{3}$$

$$\tau_{st}(v) = \frac{1}{2}(-|b_{st}(v)| + \sum_{e \ni V} |x(\vec{e})|) \tag{4}$$

Equation 3 calculates the normalized sum of current flowing through node v using $\tau_{st}(v)$ and a normalizing constant. Equation 4 defines the current flow using $b_{st}(v)$, and it ensures that the sum of all $b_{st}(v)$ is equal to 0 while considering the resistance of edges in the centrality measure. Edge resistance between nodes i and j is set according to Equation 5, allowing easier current flow and higher centrality along edges connecting nodes with higher outage costs:

$$r(\vec{e_{i,j}}) = \frac{1}{\max(c_i^{outage}, c_j^{outage})} \tag{5}$$

The weighting of the edges is updated throughout the simulation runs, incorporating experiences from previous runs into the placement strategy. Thus, the defender adjusts its sensor placement based on the previously observed effects of the attacker's actions.

F. Steering Actions

For the game-theoretic interaction between the attacker and defender, a target value is needed against which the attacker and defender can measure the yield of their actions. The expected return is calculated based on risk assessment methodologies for cyberattacks on SCADA systems, considering vulnerability, threat/attack, countermeasure, and impact [17]. Risk is a target value to be minimized from the defender's perspective, aiming to avoid or reduce potential damage, or maximized from the attacker's perspective, aiming to achieve the target. The β-TTC Metric for Practical Cyber Security Risk Estimation considers both vulnerabilities and attacker skills to estimate the time it takes to compromise a system [18]. Based on this, the TTC and the success rate of an attack against each component P_j are calculated using the model from [18]. The model requires vulnerability data from open-source datasets such as the Common Vulnerabilities and Exposures (CVE), making it compatible with MulVAL requirements.

$$TTC(s, W) = t_1 \cdot P_1 + t_2 \cdot (1 - P_1) \cdot (1 - u) \tag{6}$$

Impact is represented by the outage costs of each component C_i, calculated based on their importance to the grid operation. The components' outage costs are determined using the Purdue model [19] and the customer interruption cost of unplanned outages by peak power consumption for the industry sector from the model in [20]. A worst-case scenario of a 12-hour outage is considered. The risk is calculated using Equation 7:

$$Risk = \sum_{i=1}^{N} P_i \cdot C_i \tag{7}$$

Based on this dynamic interaction between the attacker and defender, we can simulate a logically bound chain of events within a cyber incident involving various actions. This simulation approach is then used to generate synthetic data of cyber incidents, considering flexible scenarios between the attacker and defender.

IV. Results

In this section, we present the results of the proposed training procedure for IDS based on ML using synthetic attack data.

A. Investigation Procedure

The goal of this research is to understand the impact of complex patterns in attack data and investigate the effect of different levels of complexity on data quality. To achieve this, we conducted an investigation using a multi-layer network architecture based on the Purdue model [19]. The architecture consists of 21 subnets representing a smart grid within an industrial control context. Various countermeasures of the defender and capabilities of the attacker were integrated at different levels. For example, we varied the coverage of IDS sensors that sniff corporate network traffic, directly influencing attack propagation. Using 5 to 15 sensors strikes an optimal balance between generating enough alarms from malicious activity and avoiding excessive defender intervention in our case studies. Additionally, we manipulated parameters affecting the speed and spread of the attack, enabling the generation of complex datasets. We evaluated data quality and its variation by training selected ML-based anomaly detection models on log files generated during attacks in the Unified2 IDS event format [21] (cf. Table II). These are the input features for the ML models, which aim to predict whether the alerts generated are from background processes or due to the attacker's actions based on the format of the alerts. The best performing models were selected and compared using the evaluation method. Furthermore, we analyzed the influences and expression of data quality with respect to parameterization and presented the findings accordingly. The investigation was conducted on a PC equipped with a multicore CPU, at least 16GB RAM, a dedicated GPU with 8GB+ VRAM, and fast storage, such as SSD. The investigation involved synthetic data generation, game-theoretic interactions, path optimization, and attack graph generation with $O(N^3)$ complexity.

TABLE II
GENERATED ALERTS FROM THE TOOL IN UNIFIED2 FORMAT

label	size	label	size
sensor id	4 bytes	source port/icmp type	2 bytes
event id	4 bytes	dest. port/icmp code	2 bytes
event second	4 bytes	protocol	1 byte
event microsecond	4 bytes	impact flag	1 byte
signature id	4 bytes	impact	1 byte
generator id	4 bytes	blocked	1 byte
signature revision	4 bytes	mpls label	4 bytes
classification id	4 bytes	vlan id	2 bytes
priority id	4 bytes	padding	2 bytes
ip source	16 bytes	application id	NA
ip destination	16 bytes	sequence number	NA

B. Complexity

To study the diversity of actions, we measured the complexity of the attack on a scale from 0 (non-existent) to 10 (very complex attack) using the Common Vulnerability Scoring System (CVSS) [22] and the length of the propagation path. Different start configurations were used to test ML models on 30 consecutive attacks (cf. Figure3). The attacker's skill level started at 0.5 and was incremented by 0.02 until reaching 1 after the 25th attack. We considered three different scenarios for the starting capital and the funds gained per second. The number of sensors was also varied to manipulate the difficulty

of the attacks. The complexity of the attacks was calculated based on the exploit complexity of the vulnerabilities used. Figure 3 shows the average complexity score of the exploited vulnerabilities for all 30 performed attacks with a 95% Confidence Interval (CI). Increasing the number of IDS sensors and the amount of funds led to higher attack complexity. Complexity also increased when the attacker was required to take alternative paths after reactive countermeasures were deployed or when more vulnerabilities were exploited.

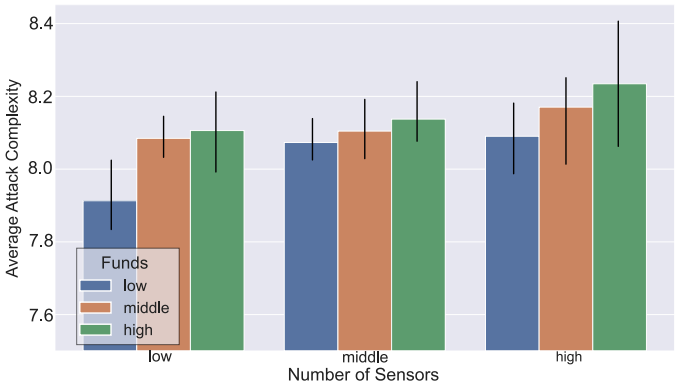

Fig. 3. Attack Complexity of the generated attacks with different start configurations. The y-axis represents the average complexity score over the simulation runs, while the error bar represents the 95% CI. The x-axis represents the number of placed IDS sensors, and the color labels of the bars represent the investment of the defender in preventive measures.

C. Performance

To select an appropriate ML algorithm for classification, we considered techniques presented in the IDS review [22] and a referenced paper [23]. From the supervised learning methods, Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), Complement Naïve Bayes (CNB), and Extreme Gradient Boosting (XGB) were chosen. CNB was preferred over the Gaussian variant due to its better performance with imbalanced datasets [24]. K-Means was used as an unsupervised learning algorithm. Based on previous research [22], [23], we chose the K-Means algorithm for time series data and used Hyper Parameter Grid Search to select the optimal value of k, considering performance and minimal overfitting observed in our experiments. The models were iteratively evaluated against the data generated by the simulation runs, with cumulative data from previous runs used for training and the current iteration for testing. The model parameters were optimized to balance overfitting and predictability. The results were presented based on a scenario with a medium number of sensors and funds. Although other scenarios were tested, the performance ranking of the models only showed minor differences. Table III reports the performance of each ML technique for the last simulation iteration (30th), including various classification metrics such as accuracy, F_1-score, Area Under Curve (AUC), and Matthews correlation coefficient (MCC), which are useful for imbalanced datasets. Notably, SVM exhibited slow fitting time with large amounts of data, as observed in [25], and was not useful for the evaluation method described in Subsection IV-A. The K-Means model was found to be the worst performing, while the XGB model performed the best based on training time and performance metrics. The performance of the RF (cf. Figure 4) and XGB (cf. Figure 5) models over simulation runs 1-29 are depicted in graphs showing the number of attacks and their success rates. Both models initially struggled with recognizing specific attacks but eventually gained an understanding of the attack structure. It is worth noting that there were no successful attacks after the 23rd one, and the attacker's skill level reached a plateau of 1 after the 25th attack. We further evaluated the defender implementation through three different methods of generating attack data (cf. Figure 6). The simulation runs were extended to 50 iterations to ensure consistency in the results. The first method involved interaction with the defender, similar to the previous section. In the second method, called "single attack", the attacker chose random paths without the defender taking countermeasures. The third method involved the attacker taking the most optimal path without any defender implementation. XGB models were trained on attack data from attacks 1-29 and tested on attacks 30-50, and each model was tested on data from the other generation methods. The results of the comparison between the three scenarios, namely "random", "single attack", and "with defender" revealed that the training data generated with the defender resulted in significantly improved detection quality. This can be attributed to the increased diversity and the presence of plausible attack patterns that were not as pronounced in the random or single attack scenarios, as the defender's countermeasures forced the attacker to alter their strategy from the optimal short path.

TABLE III
SCORES OF THE DIFFERENT ML MODELS USING VARIOUS METRICS

Metric	RF	DT	SVM	CNB	K-Means	XGB
Accuracy	0.9375	0.8182	0.9382	0.6697	0.5003	1
Recall	0.8889	0.8889	0.89	0.6810	0.0003	0.8889
Precision	0.9999	0.7273	0.9889	0.7	0.5555	1
F_1-Score	0.9411	0.8000	0.9369	0.6904	0.0006	0.9412
AUC	0.9444	0.8391	0.9251	0.6841	0.5279	0.9444
MCC	0.9333	0.6471	0.8721	0.3367	0.0005	0.9428

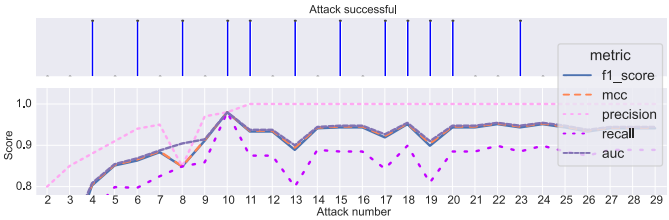

Fig. 4. Evaluation of the RF model. The x-axis represents the simulation iteration, while the color labels of the lines represent different metrics.

D. Discussion

During the investigation, we observed that increasing the number of sensors and available funds for defense led to more complex attacks. This occurred because the attacker was forced to take different, more difficult paths with complex vulnerabilities. This could happen either because the original path

Fig. 5. Evaluation of the XGB model. The x-axis represents the simulation iteration, while the color labels of the lines represent different metrics.

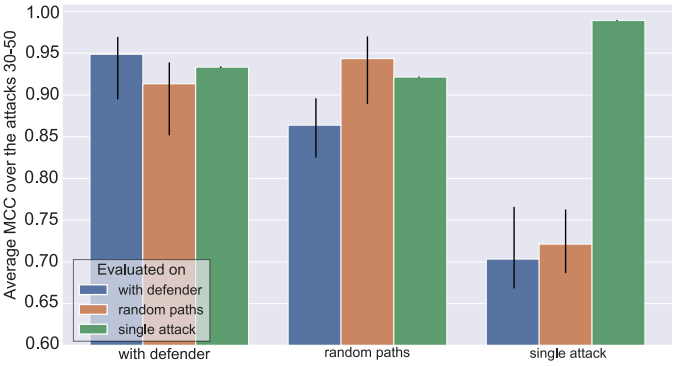

Fig. 6. Evaluation of the different generation methods. The y-axis represents the MCC over simulation runs 30-50, while the error bar represents the 95% CI. The x-axis represents the training data, and the color labels of the bars represent the test data.

was blocked by sensors and reactive countermeasures from the defender or because the attacker became discouraged by the low success rate of their actions due to increased preventative countermeasures requiring more funds. The evaluation also revealed that the predictive ability of ML models improved as more attacks were trained, providing more data for the models to learn from. The dynamic interaction between the attacker and defender generates diverse data for ML training, allowing the models to adapt and handle evolving attack vectors. Enriching the data with interaction between the defender and attacker during training resulted in better detection quality for the specific trained model. This was especially evident in the last experiment, where data involving a defender in the attacks exhibited a more intricate relationship compared to the other two methods. Models trained on other data were unable to comprehend the attack structure, while the model trained on defender-inclusive data performed better in handling diverse attack types.

V. Conclusion

This study proposes a model for generating synthetic data to train ML algorithms for intrusion detection in power grids. The approach utilizes attack tree modeling and a game-theoretic method to create realistic data. The attack simulation component generates attack trees while the defender minimizes the risk of attacks. A sensitivity analysis is conducted by adjusting the number of sensors and defense funds. The results demonstrate that RF and XGB achieve the best performance on the generated data. However, the model has certain limitations, such as the requirement for knowledge of existing rules in the attack simulation component and the potential omission of certain types of attacks in the attack-defense dynamics. Future research could explore the integration of more sophisticated ML algorithms and more realistic attack scenarios to enhance the model's accuracy and expand its scope. Additionally, investigating the model's effectiveness in detecting attacks on larger and more complex power grids would further improve its applicability. Lastly, incorporating human factors, such as social engineering tactics, into the attack-defense dynamics could provide a more comprehensive understanding of cyber threats to power grids. Nevertheless, our approach offers valuable insights to guide research in this direction.

References

[1] T. Krause *et al.*, "Cybersecurity in power grids: challenges and opportunities," *Sensors*, 2021.

[2] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," *E-ISAC*, 2016.

[3] L. Bader *et al.*, "Comprehensively analyzing the impact of cyberattacks on power grids," *Euro S&P*, 2023.

[4] J. McHugh, "Intrusion and intrusion detection," *IJIS*, 2001.

[5] I. Sharafaldin *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, 2018.

[6] C. G. Cordero *et al.*, "On generating network traffic datasets with synthetic attacks for intrusion detection," *ACM TOPS*, 2021.

[7] S. K. Pandey *et al.*, "Gan-based data generation approach for ids: Evaluation on decision tree," in *AISC: V14*, 2021.

[8] V. Babu *et al.*, "Melody: synthesized datasets for evaluating intrusion detection systems for the smart grid," in *WSC*, 2017.

[9] A. Dutta *et al.*, "Deep reinforcement learning for cyber system defense under dynamic adversarial uncertainties," *arXiv preprint arXiv:2302.01595*, 2023.

[10] D. Agnew *et al.*, "Implementation aspects of smart grids cyber-security cross-layered framework for critical infrastructure operation," *Applied Sciences*, 2022.

[11] S. Musman *et al.*, "A game theoretic approach to cyber security risk management," *JDMS*, 2018.

[12] X. Ou *et al.*, "Mulval: A logic-based network security analyzer." in *USENIX*, 2005.

[13] B. E. Strom *et al.*, "Mitre att&ck: Design and philosophy," in *Technical report*, 2018.

[14] P. E. Kaloroumakis *et al.*, "Toward a knowledge graph of cybersecurity countermeasures," *MITRE*, 2021.

[15] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, 2022.

[16] U. Brandes, *Network analysis: methodological foundations*. Springer, 2005.

[17] Y. Cherdantseva *et al.*, "A review of cyber security risk assessment methods for scada systems," *Computers & security*, 2016.

[18] A. Zieger *et al.*, "The β-time-to-compromise metric for practical cyber security risk estimation," in *IMF*. IEEE, 2018.

[19] T. J. Williams, "The purdue enterprise reference architecture," *Computers in industry*, 1994.

[20] K. Sinan *et al.*, "A novel hybrid approach to estimate customer interruption costs for industry sectors," *Engineering*, 2013.

[21] H. Au *et al.*, "Multi-stage analysis of intrusion detection logs for quick impact assessment," in *ECIW*, 2016.

[22] T. Saranya *et al.*, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science*, 2020.

[23] J. W. Teo *et al.*, "Evaluating synthetic datasets for training machine learning models to detect malicious commands," in *SmartGridComm*. IEEE, 2022.

[24] A. Kelly *et al.*, "Investigating the statistical assumptions of naïve bayes classifiers," in *CISS*. IEEE, 2021.

[25] H. Liu *et al.*, "Machine learning and deep learning methods for intrusion detection systems: A survey," *applied sciences*, 2019.