Xgboost-based Multi-steps Cybersecurity Attacks Detection Model

ABHISHEK HIREMATH[1], LATHA P H[2], CHETHAN J[3], DINESH S[4], HARSHITHA S[5]
[1, 2, 3, 4, 5]*Rajiv Gandhi Institute of Technology*

Abstract- Millions of businesses have begun to use the web in the last twenty years as an inexpensive way to connect with customers and carry out transactions with customers. Cloud-based electronic storage and information databases are commonly used on the internet. It retains information that consumers freely submit via web-based management sites, purchasing carts, logging inputs, and exploration and submitting forms. Simply, these programs, as common as they are, are highly susceptible to cyber threats and attacks that are performed by hackers. Concurrently with these advancements, developing a dependable web-based app is a challenging endeavor fraught with difficulties. Malware and other cyber threats to the privacy, security, and accessibility of networked devices are types of that threats that are challenging for web-based business sites.

I. INTRODUCTION

Intrusion Detection System (IDS) strategies can be classified into two categories: detection-based systems and monitoring- based systems. The monitoring-based method employs a pair of systems: a Network Intrusion Detection System (NIDS) and a Host Intrusion Detection System (HIDS). NIDS tracks and detects attacks in every network component, protecting every network connection, whereas HIDS is used in order to safeguard a particular device.

Because of their broad availability and significant practice, web-based applications are exposed to cyber-attacks. Attackers are frequently rendering a web-based application unreachable by sending requests that are malicious [2]. Burglars or scammers could infect a vulnerable Web implementation, compromising the safety, integrity, and accessibility of the tasking's supplies [3]. This might result in monetary losses and permanent business degradation. The dangerous attacks can harm the World Wide Web apps in a wide range of methods, involving causing resource damage, getting data from databases, halting operations, or acquiring accessibility to the implementation.

Multiple anomaly detection investigations on web-based applications that researchers have released in the scientific community. For instance, Nguyen et al. [9,10] used CSIC-2010 to introduce an overall attribute selection for identifying attacks from hackers. The selected 30 suitable attributes are used for tasking-based cyber assault categorization. A continuously evolving identification method that outperforms corresponding methods in terms of accuracy with NaiveBayes achieved 72.78%, BayesNetwork achieved 82.79%, A-ExIDS achieved about 90.98%, Decision Stump achieved about 74.73%, Hedge/Boosting achieved about 82.1%, Majority Voting achieved 81%, A-IDS achieved about 90.52%, and RBFNetwork achieves about 72.46% [11]. Tekerek et al. [12] built an integrated web app security system (firewall) to prevent internet-based cyberattacks. The authors mixed anomaly identification-based and signature recognition-based techniques to enhance the effectiveness of identification. Kozik et al. in [13] proposed an approach to identify Web-based attacks via the internet that utilized the Hypertext Transfer Protocol, or HTTP, requests headings the use of frequent expressions for categorizing cyber-attacks, alongside an accuracy rate for the detection reached

Millions of businesses have begun to use the web in the last twenty years as an inexpensive way to connect with customers and carry out transactions with customers. Cloud-based electronic storage and information databases are commonly used on the internet. It retains information that customers freely submit via web-based management sites, purchasing carts, logging inputs, and exploration and submitting forms. Simple, broad programs, on the other hand, are frequently hacked.

Because of their broad availability and significant reach, web-based applications are exposed to cyber-attacks. Attackers are frequently rendering a web-based application unreachable by sending requests that are malicious. Burglars or scammers could inject a vulnerable web implementation, compromising the safety, integrity, and accessibility of the tasking's supplies. This might result in monetary losses and permanent business damage. According to data from the World Wide Web, users in a wide range of media, including websites, software programs, and getting data from installations or operations, are acquiring access illegally to the implementation.

XSS (cross-site scripting), SQL Language Injection (SQLI), and maliciously authenticated XML are identified as web-based application safety hazards according to the OWASP (Open Web Application Security Project) Top 10 Safety Potential vulnerabilities list for the year 2021.

Intrusion Detection Systems (IDSs) can be classified into two types: IDS based on signatures and tasking-based IDS. The first type is used to analyze rule-matching or harmful behavior, which is especially effective for recognizing already-identified attacks. In contrast, tasking-based identification is a type of detection that identifies attacks whose information is out of the ordinary for the stream. Tasking-based identification is used to detect unidentified and zero-day intrusions by assessing the behavior of a framework.

Multiple anomaly detection investigations on web-based applications have been released in scientific documentation. For instance, Nguyen et al. (2019) used CSIC-2010 to introduce new attribute selection for identifying attacks from hackers. The selected 30 suitable attributes are used for tasking-based cyber-attack categorization. At a continuous evaluation, the findings reveal that their outperforming corresponding methods in terms of accuracy with NaiveBayes achieving 72.78%, BayesNetwork achieving 82.79%, AE-IDS achieving about 90.98%, Decision Stump achieving about 74.73%, Hoeffding achieving about 82.14%, Majority Voting achieving 81%, AIDS achieving about 90.52%, and RBFNetwork achieving about 72.46%.

Gao et al. (2014) developed a novel approach for forecasting attacks via the Internet. Epp et al. (2015) presented a tasking-based detection approach for web-based cyber-attacks using a Support Vector Machine (SVM) employing CSIC2010 as well as CSIC2012v2 data sets. The proposed strategy has an estimated F1 rating of 93% as well as a determined TPR (True Positive Rate) of 95%. Moreover, Wang et al. investigated the use of deep learning methods for assessing CNN, LSTM, and their ensemble approach.

II. PROBLEM STATEMENT

With the rapid growth of software and networks, cyber-attacks have increased significantly. Existing Intrusion Detection Systems (IDS) need accurate and up-to-date datasets to detect new and complex multi-step cyber- attacks effectively.

The paper addresses this challenge by using the MSCADD dataset, which contains multiple attack types, and applying machine learning algorithms (e.g., XGBoost, Random Forest) for multi-step attack detection.

The research aims to:

- Identify and classify different types of web-based and network-based attacks.
- Improve accuracy using suitable feature selection and machine learning techniques.
- Evaluate performance with metrics such as AUC and ROC.

Input
Dataset: MSCADD (Multi-Steps Cyber Attacks Dataset) containing 128,799 instances.

Features: 67 characteristics (network variables) extracted from PCAP files.

Attack Classes:

1. URL Scan – 28,502 instances (22.12%)
2. Message Scan – 10,188 instances (8.63%)
3. APK Dectiction – 88,502 instances (68.71%)
4. Spam Call Detector – 1,565 instance

Output

Classification of network traffic into normal or one of the attack categories.

Trained machine learning models that can detect and classify multi-step cyber-attacks. Performance evaluation results (accuracy, AUC-ROC).

Results

Models tested: Gaussian Naive Bayes, Decision Tree, K-Nearest Neighbors, XGBoost, Random Forest, and others.

Best performance: Achieved accuracy of 99.9% for multi-step attack detection. Metrics used: AUC, ROC, and comparison with existing works.

For future work, even yet, this strategy is not feasible in applications that operate in real-time

The model successfully distinguished between multiple types of attacks with high accuracy after feature selection and training.

III. OBJECTIVES

1. Accurate Multi-Step Attack Detection

Detect complex, sequential cyber-attacks (e.g., password cracking followed by DDoS) with high accuracy.

2. Leverage XGBoost for Classification

Use the XGBoost algorithm, along with other classifiers, to improve detection performance and robustness over traditional IDS methods.

3. Utilize the MSCADD Dataset Effectively

Employ a real-world, multi-class dataset containing various internet-based attack types such as Port Scan, Brute Force, HTTP Flood, ICMP Flood, and Web Crawling.

4. Feature Selection for Optimization

Identify and use the most relevant features to improve classification accuracy and reduce computational cost.

5. Achieve High Evaluation Metrics

Maximize performance indicators like accuracy, AUC, and ROC while maintaining low false positive rates.

6. Address Real-World IDS Challenges

Provide a scalable and reliable Intrusion Detection System capable of handling the growing volume and complexity of cyber threats.

IV. EXISTING SYSTEM

1. Uses traditional Intrusion Detection Systems (IDS) that are either signature-based (matching known attack patterns) or monitoring-based (tracking unusual network behavior).
2. Some methods use single-step detection — identifying one attack type at a time, not sequences of attacks.
3. Machine learning approaches like Adaboost and Random Forest have been applied to detect attacks using the MSCADD dataset.
4. Previous work used only 10 features from the dataset for classification.
5. Models were evaluated using metrics like AUC and InfoGain, focusing mainly on detecting single attacks.

V. DRAWBACKS

1. Cannot detect multi-step attacks well – Older systems struggle when attacks happen in a sequence (e.g., password cracking followed by DDoS).
2. Limited feature use – Using only 10 features misses important attack indicators, reducing accuracy.
3. Fewer classifiers – Previous studies tested with only a couple of machine learning models, limiting performance improvements.

4. Less robust for new threats – Signature-based methods fail when the attack is new or unknown.
5. Lower accuracy compared to potential – Performance can be improved with better feature selection, more algorithms, and enhanced preprocessing.

VI. PERCENTAGE OF WORK COMPLETED

The project has progressed significantly , with most modules fully developed and tested .The status of each component is as follows:

Section Weight in Overall Work Status (if you've only decided topic & basic concept) % Completed

1. Title & Abstract 100%
2. Introduction / Problem Statement 100%
3. Literature Review 75%
4. Objectives & Scope 85%
5. Proposed Methodology (XGBoost, multi-step attack detection flow, dataset, features) 95%
6. Expected Results / Significance 75%
7. References 85%

Timeline / Work Plan 100%

VII. POSSIBLE OUTPUT OF THE PROJECT

Message Scan (SMS / Chat Message Detection)

Functionality: Detects phishing, scam, or malicious messages using NLP features (keywords, sentiment, suspicious patterns).

Output Examples:

"safe_message" → No suspicious content detected. "phishing_message" → Contains suspicious link & scam keywords.

2. URL Scan (Malicious Link Detection)

Functionality: Analyzes URLs for phishing/malware distribution using domain reputation, WHOIS data, and lexical features.
Output Examples:

"benign_url" → Reputable domain, no phishing signs.

"malicious_url" → Domain blacklisted; URL resembles known phishing patterns.

3. APK Scan (Malware Detection in Android Apps)

Functionality: Static and dynamic APK analysis (permissions, API calls, bytecode features). Output Examples:
"benign_apk" → Normal permissions, no known malicious signatures.

"malicious_apk" → Requests excessive sensitive permissions; matches known malware family.

4. Spam Call Detector

Functionality: Detects telemarketing, scam, or robocalls using caller ID analysis, call frequency, and user reports.
Output Examples:

"normal_call" → Number not in spam lists, normal call frequency.

"spam_call" → Matches spam database; suspicious calling patterns detected

5. Multi-Step Detection Integration

Functionality: Links suspicious events across categories (e.g., a phishing message contains a malicious URL leading to an APK download → results in a spam call).
Output Example:

{ "stage_1": "phishing_message_detected", "stage_2": "malicious_url_detected", "stage_3": "malicious_apk_detected", "stage_4": "spam_call_detected",
"verdict": "multi-step_attack_detected"

}

CONCLUSION

This research investigates the implementation of algorithms for selecting features to categorize cyber-attacks. Simulations were conducted on 128799

samples from the MSCAD database, which has 67 attributes including details about most common network variables, protocols, and attacking diversity. The method of implementation includes separate processes such as pre-processing cleaning up data, partitioning data into sets to be used for training and testing, picking features, and creating classification algorithms. Finally, nine distinct classification methods such as RF, KNN, NB, DT, XGB, and CatBoost, were used for categorizing key features to recognize attacks on networks and obtain better detection outcomes. Based on their effectiveness, the performance analysis uses F1-score, Recall, Accuracy, and Precision to evaluate each technique's efficiency. Based on the obtained results, Random Forest and XBG algorithms are the most effective Web Attacks classifiers regarding the AUC. For future work, even yet, this strategy is not feasible in applications that operate in real-time. Such a problem may be solved by implementing capabilities that aggregate data from the network's sensors in an instantaneous manner and give conclusions based on Machine Learning approaches.