Online at: https://researchtrendsjournal.com
Indexed Journal

ISSN No: 2584-282X
Peer Reviewed Journal

INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

Volume 2; Issue 6; 2024; Page No. 257-261

Received: 08-08-2024
Accepted: 19-10-2024

Intrusion detection systems for SQL databases using machine learning

[1]Chadchankar Amarnath Shivanand, [2]Dr. Balveer Singh, [3]Dr. Yashpal Singh, [4]Dr. Rohita Yamaganti and [5]Dr. Swati Dey

[1]Research Scholar, P.K. University, Shivpuri, Madhya Pradesh, India
[2-5]Professor, P.K. University, Shivpuri, Madhya Pradesh, India

DOI: https://doi.org/10.5281/zenodo.17132174

Corresponding Author: Chadchankar Amarnath Shivanand

Abstract

SQL Injection is still among the worst security flaws, exposing users' private data and causing financial losses. The most current OWASP Top 10 study ranks injection attacks as the top vulnerability, and the frequency of these attacks is on the rise. By definition, Intrusion Detection System (IDS) rules that rely on static signatures are a common component of traditional defensive systems. These rules are great for preventing known attacks but aren't effective against zero-day threats. Many recent studies have used machine learning methods, which may identify previously unseen threats but can be performance-heavy depending on the methodology. To add insult to injury, some intrusion detection systems scour database server logs for information, while others focus on gathering traffic entering the web app across the network or the web app host. A web application host, a MySQL database server, and a Datiphy appliance node placed between the two are the two sources of traffic that will be collected in this project. Through our examination of these two datasets as well as an additional dataset that is correlated with them, we have proven that the accuracy achieved with the correlated dataset using algorithms like decision trees and rule-based approaches is comparable to that of a neural network algorithm, but with substantially better performance.

Keywords: Industry, traditional, machine learning, performance, companies

Introduction

Web assaults like SQL Injection have been around for a long time, but they still endanger individuals' private information as well as the security of companies and governments suffer financial losses as a result. That is especially the case when considering new attack vectors are always appearing and old ones are constantly evolving. Web attack mitigation receives substantial funding from industry and security businesses; yet, many existing mitigation solutions have limits that are being actively sought to be overcome by current research.

Static analysis of incoming online traffic, often called one typical approach to traditional online attack mitigation is signature detection. Creating a signature that is specific to cyberattacks is the essence of this strategy; when a firewall or other security appliance detects this signature, potentially obstruct the suspicious traffic. Although this method is quick and may be used in real-time to protect network assets, it does have one drawback: it can only identify known assaults.

One other way to protect against SQL injection attacks on the web is to examine the syntax of incoming SQL queries; an attack of this kind is identified when a query is found to be erroneous. Unfortunately, this technique requires extensive understanding of the program and the framework of "ordinary" questions; yet, it manages to do a fantastic job of detecting novel threats using incorrect queries and has good detection results overall.

One approach to detecting SQL injections that is now being investigation use techniques derived from machine learning. The employment of SVMs, decision trees, neural networks, and rule-based learning techniques is common in this field of study. The ability to identify novel assaults is a major strength of these methods. One possible downside of these strategies is that, depending on the algorithm utilized, processing time might potentially rise.

Intrusion Detection Systems

One of the main functions of an IDS is to keep an eye out for any suspicious or illegal behavior on a network or inside its systems. Usually, In the event that a security incident occurs, a SIEM system will either alert the administrator or compile all of the pertinent data into a single location of an incursion or breach. By combining data from many sources and using alert filtering algorithms, a SIEM system can distinguish between legal and hazardous activity.

Intrusion detection systems' many spheres of influence vary from individual PCs to extensive networks. These days, NIDSs and HIDSs are the two main categories used to describe these kinds of systems. An HIDS would be a system that keeps tabs on critical OS files, whereas an NIDS would be one that examines all incoming network traffic. Another way to categorize IDS is by detection method. Two of the most popular variations are signature-based detection and anomaly-based detection are two methods for detecting harmful patterns and abnormal traffic, respectively. The former employs machine learning to find signatures, while the latter analyses historical data to identify deviations from a model of "good" traffic. One more popular variation is reputation-based detection, which involves identifying possible threats based on their reputation ratings. The capacity to react to detected intrusions is a feature of several IDS solutions. Intrusion prevention systems (IPS) are often used to describe systems that can respond to potential intrusions. The use of specialized tools may also make intrusion detection systems more useful; for example, a honeypot can be used to detect and identify malicious traffic.

Machine Learning

Machine learning tasks

There are primarily two stages to any creating an algorithm for machine learning. Different models have different ways of being trained. Common frameworks used for this purpose and for designing machine learning models include scikit-learn, Tensor flow, PyTorch, Matlab, and Weka. When it comes to optimizing the method or the number of parameters accessible, the framework is king.

Among the various applications of machine learning models, intrusion detection finds special attention in three: classification, regression, and reconstruction. Entries are sorted into many classifications by classification, which might include "normal" or "attack" or even distinct families of assaults. In order to find continuous values, such is the possibility that a given input constitutes an assault, regression (also referred to as "prediction") is used. Lastly, only a certain class of neural networks can do reconstruction. In order to get representation learning allows the network to acquire the characteristics, this job attempts to decompress and recompress the input data in an effort to rebuild it.

Machine learning algorithms may be trained in one of two ways: supervised or unsupervised. Neural networks are one example of a model that may be taught in either direction. The majority of models are trained via supervised learning, where the dataset includes inputs and the respective correct results. Assigning these outputs to their respective inputs is a mathematical function, which the algorithm learns to represent. Classification and regression are two classic supervised training problems. Contrarily, findings from the training dataset are not used in unsupervised training. Familiarizing oneself with intriguing data structures is its goal. An example of an activity that does not need supervision is reconstruction.

Machine learning models need testing once training is complete in order to evaluate their efficacy. Data that wasn't in the training set can't be used for this assessment. If the model had seen this data before—or the right outcomes in the case of supervised learning—then the assessment would be skewed. You can also compare alternative parameters' values (for example, a learning rate or neurone count) using a validation set. Finding the value that performed the subsequent top performer on the validation set step after training. We next put the whole network through its paces on the test data. Additionally, fresh data is required for a validation set; typically, a subset set aside specifically for this purpose.

Table 1: Supervised learning vs unsupervised learning

	Supervised learning	Unsupervised learning
Process	Input and output data are given.	Only input data is given.
Input data	The machine is trained using labeled data.	The machine is not given unlabeled data.
Algorithms used	SVM, NN, Random Forest, Linear and Logistics regression, Classification trees	Different categoriez: K-means, Cluster algorithms, Hierarchical clustering, and so on.
Computational complexity	Simple.	Complex.
USA of data	Uses training data and relate input and output results.	Does not use output data.
Accuracy of results	Accurate and trustworthy.	Less accurate and trustworthy.
Real time learning	Learning is offline.	Real-time.
Number of classes	Known.	Unknown.
Main drawbacks	Big data is a challenge.	No precise information in regards to data sorting, and the output is not known.

Literature Review

Deepa Manikandan *et al.* (2024) [1] Security across all domains, including databases, networks, and the cloud, has grown in importance in real-time distributed systems since the advent of cyberspace. When faced with evolving threats, present-day intrusion detection systems (IDS) find it difficult to stay involved. The proposed model DR-DBMS (dimensionality reduction in database management systems) combines supervised machine learning methods, dimensionality reduction approaches, and sophisticated rule-based classifiers to enhance the accuracy of intrusion detection for different kinds of assaults. Simulation findings show that the DR-DBMS system effectively used methods for dimensionality reduction and traits selection for determining intrusion assault in 0.07 seconds with a lesser number of characteristics.

Roland Plaka (2021) [2] Intrusion detection systems (IDS) are designed to keep an eye out for any breaches in a network's confidentiality, integrity, or availability that may have occurred as a result of malicious or unapproved activity. Machine learning methods for detection and classification, a variety of intrusion detection systems (IDS), and anomaly detection approaches were thoroughly reviewed in this thesis. We provide an architecture for intrusion detection systems (IDS) that combines traditional methods with machine learning techniques. The current detection methods may be enhanced for better attack detection and categorization by including suitable machine learning techniques. Additionally, we have made an effort to evaluate and execute a battery of virtual tests on each machine learning algorithm to compare their performance. For general intrusion detection systems used in industrial control applications, our method offers indications for choosing machine learning techniques.

Yasmeen S. Almutairi *et al*. (2022) [3] Safeguarding sophisticated communication networks requires intrusion detection systems (IDS). Certain patterns, signatures, and rule breaches were the primary targets of these systems' design. Potential new methods to network intrusion detection have emerged Using Deep Learning and Machine Learning methods in the last many years. Techniques like this may tell the difference between regular patterns and those that aren't. Support Vector Machine, J48, Random Forest, and Naïve Bytes were among the machine learning techniques used to assess the Network Intrusion Detection Systems (NIDS) in this article. The methods used binary and multi-class categorization on the NSL-KDD benchmark data set. Extensive discussion is provided about the outcomes of using such strategies, which exceeded our earlier efforts.

Amit Singh *et al*. (2024) [4] Intrusion detection systems (IDS) are facing new hurdles due to encrypted data, new protocol variety, and an increase in the amount of criminal actions globally. Intrusion detection systems that rely on signatures are now at a stage where inadequate in this case. Several academics have put forth IDSs that use machine learning to detect intrusions previously unseen harmful actions by analysing patterns of activity. Intrusion detection systems that use on machine learning have many advantages than SIDS that rely on signatures are more effective in spotting novel forms of network-based malware. This study evaluated network intrusion detection systems utilising the IDS dataset, which contains the most recent common attacks, and two data resampling techniques in addition to 10 machine learning classifiers. The top three IDS models-XGBoost, KNneighbors, and AdaBoost-perform better than binary-class classification with 99.49%, 99.14%, and 98.75% accuracy, respectively. XGBoost, KNneighbors, and GaussianNB are the most accurate in multi-class classification with 99.30%, 98.88%, and 96.66% accuracy, respectively.

Tahsinur Rahman (2022) [5] The need of security measures to prevent breaches is growing in direct correlation with the proliferation of internet-connected devices. The purpose of an Intrusion Detection System (IDS) is to identify and block potentially harmful network traffic. The need for anomaly-based intrusion detection systems that use machine learning to identify more recent assaults has arisen because new threats may easily bypass standard signature-based IDS. This thesis will focus on a specific theme: anomaly-based intrusion detection systems that use deep learning. A comparison is made between adversarial machine learning methods and Generative Adversarial Networks (GAN), with conventional deep learning techniques. We use statistical metrics on two separate datasets to assess the strategies. As part of the assessment process, malicious samples are taken into account alongside benign and known attack samples. The last step is to compare the optimal method to other open-source anomaly-based intrusion detection systems. Outperforming all other techniques was a method that used a GAN to generate adversarial samples. Furthermore, when faced with malicious data points, the method matches the performance of current anomaly-based IDS. After reviewing the literature, we have come to the conclusion that intrusion detection systems based on GANs may be enhanced to better withstand both new and malicious assaults.

Research Methodology

To determine whether incoming communication is benign or malicious, our proposed method in this study employs machine learning methods. A bespoke business chat web app running on a distant MySQL server forms the backbone of the system. There are two locations where data is recorded: first, in the HTTP traffic that flows between the servers that generate traffic and those that host the web applications. Secondly, in the data transfer between the webapp server and the offline database server using MySQL.

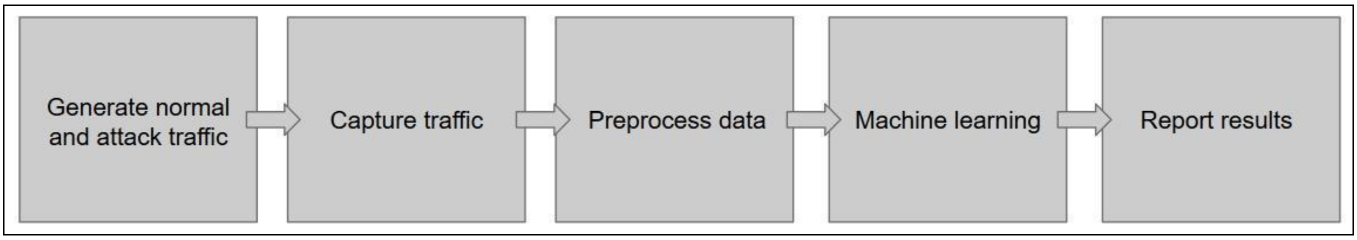

Fig 1: System Process

SCADA Dataset

After introducing the issue and ML methods utilized this section of the chapter aims to provide a general outline of the first dataset. This data collection originates from a CI water system that is controlled by SCADA.

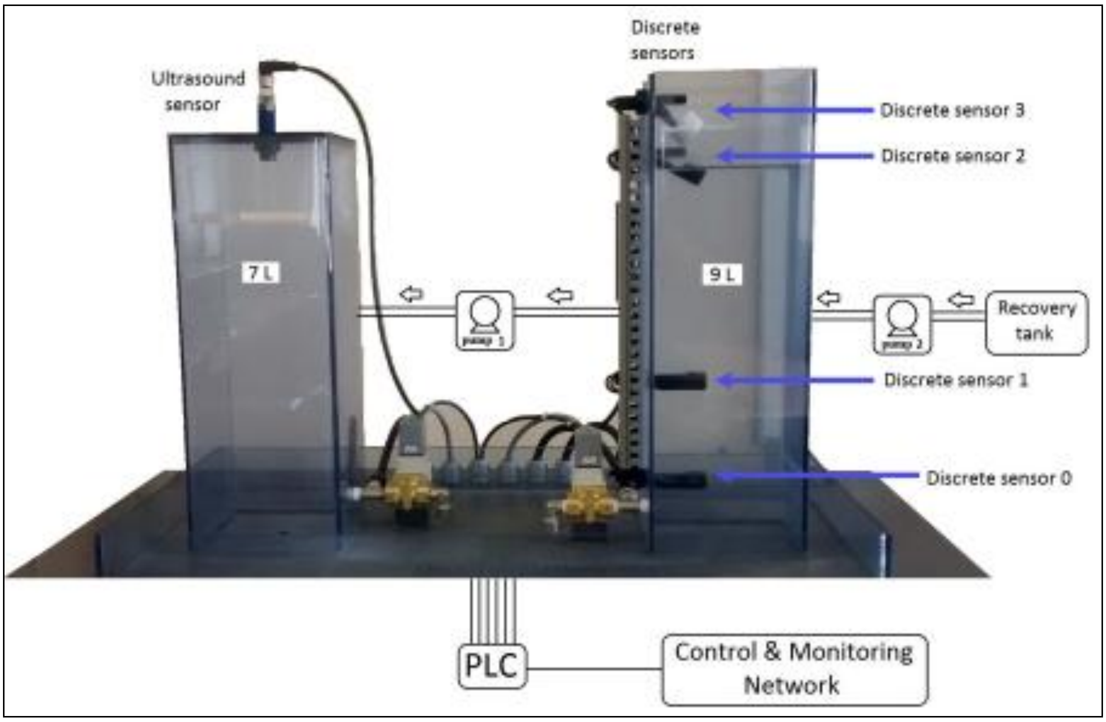

Fig 2: SCADA System Architecture

Experiment and Results

Table 1 summarizes the results of our research with several machine learning techniques. The time it takes to construct the machine learning models is called Model Time, the time it takes to classify the testing dataset using 5-fold cross-validation is called Testing Time, and Accuracy is the classification accuracy that each method achieves. Figure 1 displays the classification accuracy, whereas Table 2 displays the F-scores.

Table 2: Results With 20000 Records

Dataset	Algorithm	Accuracy	Model	Testing
			Time	Time
Webapp	JRip	94.740%	3m30.85s	2.70s
	J48	95.630%	1m42.65s	2.55s
	RF	96.525%	5m30.20s	30.60s
	SVM	94.025%	2m35.80s	1m10.45s
	ANN	96.715%	46m03.55s	3.35s
Datiphy	JRip	95.980%	6m10.90s	3.35s
	J48	96.995%	1m59.30s	2.45s
	RF	97.210%	6m10.20s	33.50s
	SVM	95.190%	2m11.50s	1m3.45s
	ANN	97.285%	41m23.00s	2.95s
Correlated	JRip	97.150%	11m50.80s	2.10s
	J48	97.295%	2m01.80s	2.05s
	RF	98.055%	4m22.55s	35.45s
	SVM	95.715%	3m30.85s	1m5.90s
	ANN	97.615%	47m25.25s	3.80s

Scada

To demonstrate the precision of anomaly detection, this section details and assesses three separate studies. The purpose of the various tests is to offer varying degrees of detail about the presence of an abnormality. From just noting that an abnormality has occurred to pinpointing the offending part and the unusual circumstance, there is a wide range of possible outcomes.

Table 3: SCADA Results: Experiment 1 - Anomaly Detection (5-fold cross-validation)

Classification (Is Anomaly)	Recall	Precision	F1-Scpre
LR			
Benign	7.15%	90.34%	13.22%
Anomaly Weighted Average	99.89%	87.7%	93.4%
	87.73%	88.05%	88.05%
NB			
Benign Anomaly	99.95%	16.74%	28.67%
	24.99%	99.97%	39.98%
Weighted Average	34.82%	89.06%	89.06%
k-NN			
Benign	74.01%	79.7%	76.74%
Anomaly	97.15%	96.12%	96.63%
Weighted Average	94.12%	93.97%	93.97%
SVM			
Benign	7.15%	92.24%	13.23%
Anomaly	99.91%	87.70%	93.41%
Weighted Average	87.75%	88.30%	88.30%
Kernel SVM			
Benign	39.53%	98.52%	56.40%
Anomaly	99.91%	91.63%	95.59%
Weighted Average	91.99%	92.545£	92.54%
DT			
Benign	74.01%	74.72%	74.35%
Anomaly	96.22%	96.09%	96.15%
Weighted Average	93.30%	93.28%	93.28%
RF			
Benign	75.66%	75.99%	75.78%
Anomaly	96.38%	96.33%	96.36%
Weighted Average	93.67%	93.67%	93.67%

MQTT: We use the six ML algorithms-LR, Gaussian NB, k-NN, SVM, DT, and RF-discussed previously to evaluate various ML approaches on the MQTT-IoT-IDS2020 dataset. The following characteristics are disabled to prevent any identifying data from being impacted: protocol, MQTT flags, source and destination IP addresses.

Table 4: MQTT-IoT-IDS2020: Overall Detection Accuracy

	Packet	Features Unidirectional	Bidirectional
LR	78.87%	98.23%	99.44%
k-NN	69.13%	99.68%	99.9%
DT	88.55%	99.96%	99.95%
RF SVM (RBF	65.39%	99.98%	99.97%
Kernel)	77A%	97.96%	96.61%
NB SVM (Linear	81.15%	78%	9755%
Kernel)	66.69%	82.6%	98.5%

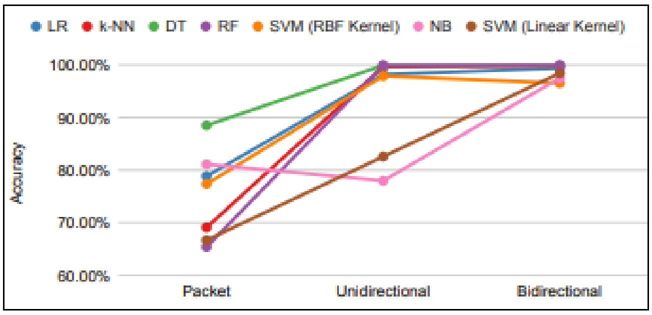

Fig 3: MQTT-IoT-IDS2020: Overall Detection Accuracy Trend using Different ML Techniques

Conclusion

Security of sensitive information, including financial and health records, remains a top concern due to SQL injection attacks and other web-based assaults. This problem is becoming more pressing as more and more social activities rely on the internet. We have demonstrated that the algorithms we have tested, including rule-based and decision tree algorithms, can classify testing data significantly faster and with accuracy comparable to that of Neural Networks, and we have also suggested a multi-source data analysis system to improve the accuracy of SQL injection attack detection. Adapting this system to detect other types of web-based attacks is on the list of future works. Other things on the list include gathering more data, like traffic going outbound from the web app to the browser, collecting larger datasets to see if that helps performance, and analyzing additional machine learning techniques for accuracy and performance.

Intrusion Detection Systems (IDS) are programs that scan every incoming and outgoing data packets for signs of malicious activity. In the last ten years, IDS have been built using a variety of ML approaches. The rapid emergence of new cyber threats has led to the widespread usage of ML approaches. This thesis delves into the use of ML approaches to construct IDS with specific purposes. In addition, this thesis explores the possibility of enhancing the efficiency and efficacy of next-generation IDS by using innovative DL approaches that have been successfully used in other areas of study.

Reference

1. Manikandan D, *et al*. Machine learning approach for intrusion detection system using dimensionality reduction. International Journal of Information Technology. 2022;34(1):1-7.
2. Plaka R, *et al*. Intrusion detection using machine learning for industrial control systems. Journal of Industrial Information Integration. 2021;24:100223.
3. Almutairi Y, Alhazmi B, Munshi A. Network intrusion detection using machine learning techniques. Advances in Science and Technology Research Journal. 2022;16(3):193-206. doi:10.12913/22998624/149934.
4. Singh A, Prakash J, Kumar G, Jain P, Ambati L. Intrusion detection system. Journal of Database Management. 2024;35(2). doi:10.4018/JDM.338276.
5. Rahman T. Intrusion detection system based on deep learning. 2022 Jul 29. [Thesis/Report].
6. Sama L. Network intrusion detection using deep learning. [Master's thesis]. Melbourne: Victoria University; c2022.
7. Kültür E. Network intrusion detection with a deep learning approach. 2022 Feb. 99 p. [Thesis/Report].
8. Worku G/Michael K. A predictive model of network intrusion detection systems using machine learning approach. 2023 Jan. [Report].
9. Raka M, et al. Intrusion detection using machine learning and log analysis. International Journal of Advanced Research in Computer and Communication Engineering. 2019;8(4):220-225.
10. Vanin P, Newe T, Dhirani LL, O'Connell E, O'Shea D, Lee B, Rao M. A study of network intrusion detection systems using artificial intelligence/machine learning. Applied Sciences. 2022;12(22):11752. doi:10.3390/app122211752.