iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60026

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Cyberattack Analysis, Detection and Prevention using Machine Learning

Gayatri Dattatraya Ranjane[1], Vishwa Hiteshkumar Joshi[2], Priyanshi Lokesh Kumar Singhal[3]
School of Technology Management and Engineering Narsee Monjee Institute of Management Studies University Navi Mumbai, India

Abstract: ***In today's linked world, with the increase of cyberattacks, it is critical to have strong detection and prevention systems. This study presents an advanced approach utilizing both machine learning and deep learning algorithms for cyberattack detection and prevention. The UNSW-NB15 dataset, renowned for its comprehensive representation of diverse cyber threats, serves as the foundation for experimentation and evaluation. Several algorithms such as Random Forest, Naïve Bayes, boosting algorithms, Artificial neural networks, Support vector machine are employed where the comparative analysis focuses on evaluating the efficiency of each algorithm in terms of recall, precision, and accuracy metrics. This study enhances the development of cybersecurity defense tactics by offering valuable perspectives on the efficacy of different machine learning methods in predicting cyberattacks. Experimental findings indicate that the boosting algorithm strategy is capable of identifying and preventing cyber threats with an accuracy rate of 94%.***

Keywords: ANN, CatBoost, Deep Learning, MLP, Naïve Bayes, Random Forest, SVM, XGBoost

I. INTRODUCTION

Cyberattacks are harmful operations carried out with the goal of compromising, disrupting, or damaging computer systems, networks, or digital devices. These assaults can target many parts of the digital infrastructure, such as software, hardware, data, and people. Cyber-attacks pose significant threats to individuals, businesses, governments, and critical infrastructure, leading to financial losses, data breaches, privacy violations, and operational disruptions (Ross, 2018). At their core, cyberattacks seek to exploit vulnerabilities in the digital ecosystem, whether they be in software, hardware, or human behaviour. As connectivity increased and digital technologies became more integral to daily life, so did the prevalence and sophistication of cyberattacks (De & Sodhi, 2020). What began as isolated incidents of mischief or curiosity has evolved into a global industry of cybercrime, espionage, and warfare, with perpetrators ranging from lone hackers to well-funded criminal syndicates and nation-states (Ashlam et al., 2022). The methods employed in cyberattacks are as diverse as the attackers themselves, encompassing a wide range of techniques and tactics. These may include exploiting known software vulnerabilities, social engineering techniques to manipulate human behaviour, or conducting widespread denial-of-service (DDoS) attacks to overwhelm target networks. As technology evolves, so do the methods of attack, with attackers constantly innovating and adapting to circumvent existing defences.

The nine types of cyber-attacks, providing a comprehensive overview for research purposes. Each attack type is succinctly defined, offering clarity and precision essential for academic inquiry. Fuzzing, Analysis attacks, Backdoors, DoS attacks, DDoS attacks, Exploits, Generic attacks, Reconnaissance attacks, and Shellcode are all elucidated, covering a broad spectrum of cyber threats. These cyber dangers highlight the significance of strong cybersecurity measures and proactive defensive methods in protecting against criminal intrusions into digital environments. Given these problems, it is critical that we take a proactive, multifaceted approach to cybersecurity. This includes investing in strong defense measures like firewalls, intrusion detection systems, and encryption technologies to protect against attackers.

II. LITERATURE SURVEY

Cybersecurity in modern technological landscapes is paramount, especially given the increasing sophistication of cyber-attacks. Among these threats, SQL injection attacks (SQLIA) provide a severe risk to online applications, databases, and network infrastructure (Gogoi et al., 2021). Traditional defense mechanisms often fall short in detecting and preventing such attacks, necessitating the exploration of novel approaches rooted in machine learning (ML) and natural language processing (NLP) techniques (Sanjeev Agrawal Global Educational (SAGE) University et al.). Table I presents a comprehensive literature overview on various methodologies proposed to address SQLIA, ranging from ensemble classification methods to deep learning frameworks, and evaluates their effectiveness in enhancing cybersecurity.

TABLE I. LITERATURE SURVEY

Research Paper	Methods Used	Dataset	Accuracy
Detection of vulnerability scanning attack using machine learning (Shahrivar, 2022)	Random Forest Algorithm, Decision Tree, BurpSuite & OWASP ZAP generate AppSensor Data	KDD'99 & NSL-KDD	Using Random Forest Precision - 98% , Recall - 96% F1 score - 0.97%
Multiphase algorithm framework to prevent SQL injection attacks using Improved ML and DL to enhance data base security in real time (Ashlam et al., 2022)	K-Means, Multi-Phase Algorithmic Framework	SQLi attack payloads, National Vulnerability database, GitHub, Kaggle and python library named Lib-Injection.	95%
Application Security using SQL Malware Detection and Prevention Scheme (Sanjeev Agrawal Global Educational (SAGE) University et al.,)	SVM Algorithm CNN for classifying SQL dataset	Made their own Dataset which consists of SQL malware scripts.	96.2% Accuracy for both Detection and Prevention of Malware Queries

Detection and Prevention of SQLI attacks and developing comprehensive frameworks using machine learning and hybrid techniques (Demilie & Deriba, 2022)	Linear and Non-linear Support Vector Machines, Natural Language Processing using word2vec	Web application inputs generated using probabilistic methods, SQLIA attack payloads generated using penetration testing	99.9%
An SQL Injection Detection Model Using Chi-Square with Classification Techniques (Adebiyi et al., 2021)	Chi-Square, Naïve Bayes Classifier, Decision Tree, K – Nearest Neighbors	KDD Test Dataset	Using Decision Tree: 98.11%
SQL Injection Attack Detection and Prevention Techniques Using Deep Learning (Chen et al., 2021)	Convolutional Neural Network, Multi-level perceptron	Training data - 25487 samples of SQL injection from the internet as negative examples, 24500 samples of the normal HTTP request as positive examples.	Accuracy-98% Precision-97% Recall - 99% F1 score - 98%
A Detection and Prevention Technique on SQL Injection Attacks (Su et al.)	Query token detection with reserved words-based lexicon	Vulnerable Query Statements	Successful Prevention from various SQLIA techniques

Cyber Attack Detection Model (CADM) based on Machine Learning Approach (Hossain et al., 2021)	Random Forest, Gradient Boosting algorithm, KNN, Decision Tree & Logistics Regression	NSL-KDD, KDD Cup 99, UNSW-NB15, URL 2016 & CICIDS 2017	Random Forest-99.90% Gradient Boosting-99.89%

A. Cyber-attack Detection Methodologies Across Different Domains

The study by (De & Sodhi,2020) focuses on utilizing machine learning techniques to detect vulnerability scanning attacks using real-world data from tCell, a web application firewall. Although achieving high precision and recall rates, the study highlights the need for improved model calibration. (Hossain et al., 2021)'s research introduces a Cyber Attack Detection Model (CADM) that employs ensemble classification methods to analyze network data patterns and bifurgate cyber-attacks. The study on UNSW-NB15 dataset demonstrates the efficacy of LASSO for feature extraction and Gradient Boosting and Random Forest algorithms for classification, enhancing attack-wise detection accuracy (Hossain et al., 2021).

In the context of cyber-physical systems, (Dānishgāh-i Shahīd Bihishtī & Institute of Electrical and Electronics Engineers) investigates the detection of deception attacks, particularly those involving false data injection. The proposed method utilizes deep neural networks to detect attacks in the early stages and resilient control algorithms for isolating misbehaving agents, showcasing improved performance over traditional methods. (De & Sodhi,2020) proposes a framework in smart grids, statistically engineered cyberattacks include random attacks, denial of service (DoS), and bogus data injection attacks. The Cyber Attack Detector (CAD) utilizes statistical coefficients and demonstrates effectiveness through experimental assessments on the Real Time Digital Simulator (RTDS). (Shahrivar, 2022) ran multiple vulnerability scans against an insecure web application, WebGoat. Next, the scanning data was trained and tested against Random Forest algorithm to test its detection accuracy. (Chen et al., 2021) on the basis of extensive domestic and international research, offered a SQL injection detection method that does not rely on a background rule base and instead use a natural language processing model and a deep learning framework, therefore lowering the false alarm rate.

B. Innovations in Cyber-attack Detection and Prevention

With the proliferation of web applications, SQL injection emerges as a common threat, exploiting vulnerabilities in database systems. (Alsahafi,2022)'s study reviews machine learning classifiers for SQLIA mitigation, it highlights the effectiveness of defensive coding approaches and the significance of performance evaluation metrics beyond accuracy. (Ashlam et al., 2022) presents an approach to SQLIA detection using NLP and machine learning, achieving high precision, recall, and f1-score. The study underscores the significance of leveraging NLP techniques for enhancing detection capabilities in web applications. (Ashlam et al., 2022) proposes a Multi-Phase algorithmic framework employing machine learning and deep learning to enhance database security against SQLIA.

Through real-world testing, (Ross, 2018)’s method demonstrates prevention of SQLI, classification of attack types, and overall database security improvement. (Demilie & Deriba, 2022) introduces a mechanism utilizing Support Vector Machine (SVM) for recognizing and preventing SQL malware queries. The proposed model achieves high accuracy in detecting and preventing SQL malware attacks, enhancing overall system security. (Adebiyi et al., 2021) has discovered a powerful machine learning solution for SQLIA mitigation, which is being implemented utilizing a defensive coding strategy based on Chi-Square. (Ahmed & Uddin, 2020) uses NLP and Bag-of-Words model to extract features and generate patterns and then integrates them with Random Forest algorithm to achieve higher accuracy.

In summary, this literature review highlights the evolving landscape of techniques for detecting and preventing SQL injection attacks, with a focus on integrating machine learning, deep learning, and natural language processing methodologies. These methodologies exhibit promising results in enhancing cybersecurity across various domains, including web applications, cyber-physical systems, and smart grids. Further our project research is warranted to address challenges such as model calibration, performance evaluation metrics, and real-time implementation to fortify defence mechanisms against cyber-attacks effectively.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

III. METHODOLOGY

A. System Architecture

An Intrusion Detection System (IDS) leveraging machine learning (ML) technologies is structured as a comprehensive system architecture in Figure 1. The process begins with the collection and preprocessing of network traffic data using the TCPDUMP. These tools capture packets traversing the network, including headers and payloads, which can provide valuable insights into network

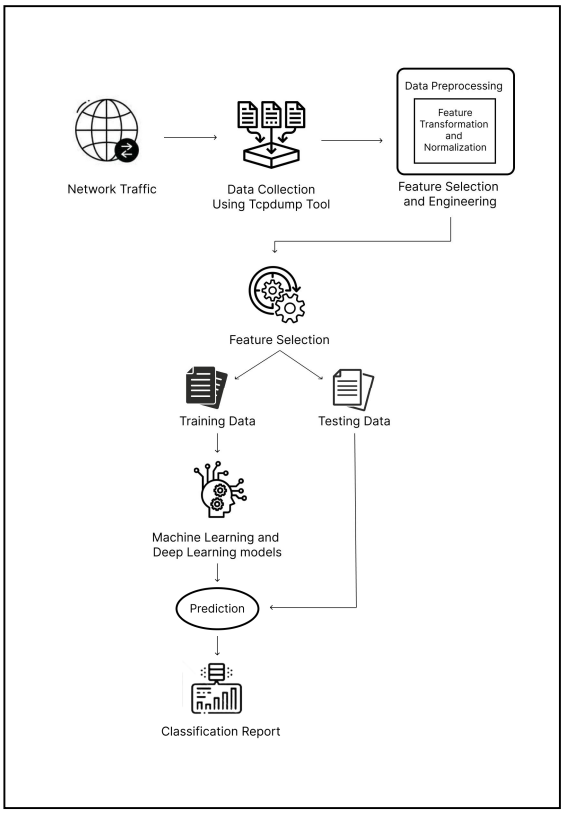


Fig. 1 System Architecture

behaviour. Raw network traffic data often contains noise and irrelevant information. Filtering, normalization, and feature scaling, which are preprocessing techniques, are applied to clean and prepare the data for analysis. The selected features comprise of packet size, protocol type, source and destination IP addresses, port numbers, and traffic patterns.

Machine learning models such as Random Forests, Support Vector Machines, and Naive Bayes classifiers are trained on labelled data to discern patterns of normal and malicious behaviour, while deep learning models like Artificial Neural Networks (ANNs) and Multilayer Perceptron (MLP) automatically learn hierarchical representations from raw data. These models are trained on labelled data, where instances are tagged as either normal or malicious traffic. Ensemble learning techniques, such as boosting, were employed to merge the predictions of multiple ML and DL models for improved accuracy and robustness. Ensemble approaches have the potential to improve upon the shortcomings of individual classifiers and produce more dependable outcomes by capitalising on the diversity of models. The IDS performance is evaluated using classification reports that provide metrics like accuracy, recall, precision, and F1-score. These metrics give insight into how well the IDS can correctly classify instances of normal and malicious traffic, as well as the trade-offs between different types of errors.

B. Dataset

The UNSW-NB15 dataset is a crucial tool for researchers studying cybersecurity because it offers a diverse range of network traffic information that captures the nuances of real-world and artificial security situations. In the controlled setting of the Cyber Range Lab at the Australian Centre for Cyber Security (ACCS), state-of-the-art tools like the IXIA The dataset was produced using the Perfect Storm program. It encompasses nine discrete attack categories, each of which represents a different aspect of possible cyber threats, ranging from denial-of-service attacks to reconnaissance. The distribution of these attacks is shown in Figure 2.

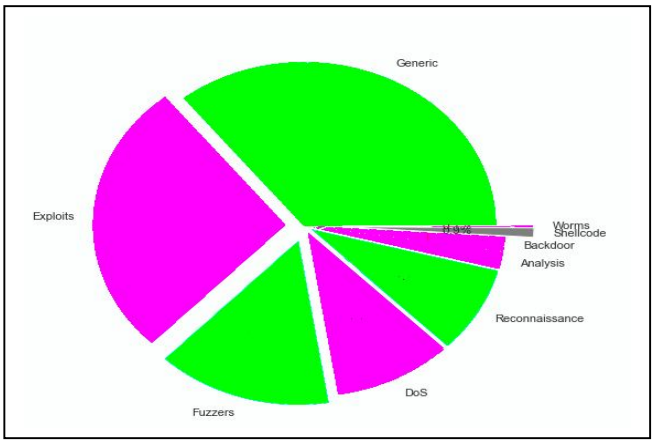


Fig. 2 Types of cyber attacks

Fuzzing is a technique which is used to find bugs and vulnerabilities in the software by providing null, or random data to trigger sudden or unlikely behavior while analysis attack involves analyzing different system or network elements to collect information. Backdoors are used to gain unauthorized access to a system which are installed by attackers to get future access. Denial of Service attacks are intended to disrupt the availability of a service, server, or network resource. Exploits take use of software, system, or network flaws to obtain unauthorized access to the system. Generic attacks are broad category of attacks that don't fit neatly into specific classifications which includes novel attack techniques, hybrid attacks, or attacks that have characteristics of multiple types. Reconnaissance gathers information to target potential vulnerabilities in a system or network. Shellcode is malicious code crafted to be executed directly by an operating system's shell. Unlike viruses, worms don't require user interaction to reproduce, making them capable of causing widespread damage or disruption quickly. The dataset's size and complexity are highlighted by the TCPDUMP tool to gather a sizable 100GB of raw network packets, which makes it a perfect resource for thorough research projects. Following that, 49 features are extracted through twelve complex algorithms and the Argus and Bro-IDS tools. These features all contribute to a more comprehensive understanding of network behaviors and security incidents. The source and destination's packet counts and bits are some of the important factors influencing the results.

The "UNSW-NB15_training-set.csv" with 175,341 records and "UNSW-NB15_testing-set.csv" with 82,332 records are developed to make model training and evaluation convenient. We utilized these datasets and applied various machine learning methods for our research.

C. Data Preprocessing

To avoid extraneous information, the training and testing datasets are first loaded from CSV files, concatenated, and the 'id' column is removed. Then, in order to make categorical variables ('proto,' 'service,' and 'state') compatible with machine learning techniques, they are encoded using a category code representation. Additionally encoded into category data is the 'attack_cat' column, which represents the attack categories. The next step is to compute a correlation matrix in order to determine which variables are highly connected. To prevent multicollinearity problems, variables having a correlation value greater than 0.95, as displayed in Figure 3 are eliminated as they are deemed highly correlated.

```
spkts sbytes 0.9643932171558754
spkts sloss 0.9718594116511157
dpkts dbytes 0.9734453279792993
dpkts dloss 0.9796120849360007
sbytes sloss 0.995771577240686
dbytes dloss 0.9967111338314709
swin dwin 0.9804584289136614
ct_srv_src ct_dst_src_ltm 0.9539519771685916
ct_srv_src ct_srv_dst 0.9794668130725328
ct_dst_ltm ct_src_dport_ltm 0.9615176986568873
ct_dst_src_ltm ct_srv_dst 0.960321213106398
is_ftp_login ct_ftp_cmd 0.9988554882935945
```

Fig.3 Variables with high correlation

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

The 'attack_cat' and 'label' columns are removed from the features, and the dataset is then divided into features (X) and labels (y). Following the aforementioned procedures for dataset cleaning, the 35 retrieved characteristics, shown in Figure 4, were employed in several algorithmic constructions of predictive models. These 35 feature's dataset overviews are displayed in Figure 4. To facilitate the assessment of the model's effectiveness, this dataset is further split using a 70/30 split for the testing set and the train_test_split function from scikit-learn. All things considered; these preprocessing procedures are essential to getting the data ready in an organized manner for further machine learning modeling.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 257673 entries, 0 to 257672
Data columns (total 33 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   dur                257673 non-null  float64
 1   proto              257673 non-null  int16
 2   service            257673 non-null  int8
 3   state              257673 non-null  int8
 4   spkts              257673 non-null  int64
 5   dpkts              257673 non-null  int64
 6   rate               257673 non-null  float64
 7   sttl               257673 non-null  int64
 8   dttl               257673 non-null  int64
 9   sload              257673 non-null  float64
 10  dload              257673 non-null  float64
 11  sinpkt             257673 non-null  float64
 12  dinpkt             257673 non-null  float64
 13  sjit               257673 non-null  float64
 14  djit               257673 non-null  float64
 15  swin               257673 non-null  int64
 16  stcpb              257673 non-null  int64
 17  dtcpb              257673 non-null  int64
 18  tcprtt             257673 non-null  float64
 19  synack             257673 non-null  float64
 20  ackdat             257673 non-null  float64
 21  smean              257673 non-null  int64
 22  dmean              257673 non-null  int64
 23  trans_depth        257673 non-null  int64
 24  response_body_len  257673 non-null  int64
 25  ct_srv_src         257673 non-null  int64
 26  ct_state_ttl       257673 non-null  int64
 27  ct_dst_ltm         257673 non-null  int64
 28  ct_dst_sport_ltm   257673 non-null  int64
 29  is_ftp_login       257673 non-null  int64
 30  ct_flw_http_mthd   257673 non-null  int64
 31  ct_src_ltm         257673 non-null  int64
 32  is_sm_ips_ports    257673 non-null  int64
dtypes: float64(11), int16(1), int64(19), int8(2)
```

Fig. 4 Independent Features

D. Models Configuration

Many machine learning models were assessed using different approaches and parameters. Their hyperparameters are shown in Table II, which are set in order to yield higher classification results from each of the models. Naïve Bayes is a probabilistic classification technique based on the Bayes theorem. Cross-validation for Naive Bayes is a technique that divides the dataset into subsets, allowing for iterative training and testing to determine the classifier's performance and generalization. Random forest was constructed using a decision tree ensemble that divided nodes based on Gini entropy. The performance of the support vector machine, which is used for both classification and regression tasks in a space with multiple dimensions, varied depending on the kernel utilized. The model used a Bagging Classifier with a Support Vector Classifier (SVC) kernel, together with ten estimators, to improve prediction speed and accuracy. Bagging, a common ensemble learning technique, enabled many models to be trained on various subsets of training data and then combined to increase overall accuracy and reduce variation. The SVC kernel, known for its effectiveness in high-dimensional spaces, provided a strong foundation for classification tasks. Additionally, the model utilized gradient boosting with decision trees as base learners to further enhance dataset performance. Gradient boosting sequentially adds weak learners, such as decision trees, to rectify errors made by previously built models, thereby improving predictive accuracy. The model also leveraged XGBoost, an optimized implementation of gradient boosting, with a softmax goal and a maximum depth of 10. By setting the softmax goal, the model was trained for multi-class classification, outputting a probability distribution over multiple class. The maximum depth parameter controlled the complexity of the decision trees, preventing overfitting.

With GPU adjusted, CatBoost, designed to handle categorical features usually, achieved comparable performance metrics to those of XGBoost. Overall, the combination of ensemble techniques, parameter optimization, and the use of powerful classifiers contributed to both the speed and accuracy enhancements of the model. Using the Rectified Linear Unit (ReLU) activation function, the multilayer perceptron, used for complex non-linear mapping, was set up with two hidden layers fully connected to each other of fifty neurons each. Moving towards more deep learning approaches, artificial neural networks, which computed a structural and functional network of interconnected neurons encompassed 22 layers and shortened regular kernel initialization.

TABLE II. MODEL HYPERPARAMETERS

Model	Parameters
Naïve Bayes	cross_validation = 5, n_jobs = -1
Random Forest	criterion: ['gini', 'entropy'], max_depth: [2, 4], min_samples_split: [2, 4] min_samples_leaf: [1, 2], random_state = 11, cross_validation = 5
Support Vector Machine	kernel='linear', probability=True, class_weight='balanced', n_estimators = 10
	kernel='rbf',gamma=0.5, C=1000.0, probability=True, class_weight='balanced', n_estimators = 10
	kernel='poly', degree=5, probability=True, class_weight='balanced', n_estimators = 10
XGBoost	max_depth: 10, objective: 'multi:softmax', num_class: 2, n_gpus: -1
CatBoost	task_type='GPU'
MultiLayer Perceptron	hidden_layer_sizes=(50,50), max_iter = 100, activation = 'relu', solver = 'adam'
Artificial Neural Network	layers = 22, kernel_initializer="truncated_normal", activation = 'relu', output_activation = 'sigmoid', cross_validation = 30

IV. METHODOLOGY

Table III compares the effectiveness of various machine learning models for cyberattack prediction using metrics of a confusion matrix. A confusion matrix summarizes how a machine learning model performed on a set of test data. True indicates that our forecasts match the actuals. True Negative (TN) indicates that both predictions and actual are of the negative class. Similarly, a True Positive (TP) occurs when both forecasts and actuals are of the positive type. A False Positive (FP) is when a prediction is positive but the actual result is negative. False Negative (FN), like False Positive, occurs when a prediction is negative but the actual result is positive. Table III infers that the recall of all models is higher than 90%.

TABLE III. CLASSIFICATION REPORT

Model	Evaluation (%)		
	Accuracy	*Recall*	*Precision*
Naïve Bayes	81.37	96.2	78.93
Random Forest	93.51	95.71	96.47
Support Vector Machine: Linear	89.58	93.53	89.53

Model	Evaluation (%)		
	Accuracy	*Recall*	*Precision*
Kernel			
Support Vector Machine: RBF Kernel	91.16	93.98	91.03
Support Vector Machine: Poly Kernel	69.32	99.98	64.28
XGBoost	94	96	95
CatBoost	94	95	95
MultiLayer Perceptron	93	93.94	94.78
Artificial Neural Network	91.79	95.9	91.69

The accuracy of a model indicates its total correctness rate. Its formula is given in (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

The precision of a model indicates the accuracy with which it predicts the target class. Its formula is given in (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall indicates if a model is able to locate every object in the target class. Its formula is given in (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

From the graphical model comparison, as seen in Figure 5, despite having a lesser precision, Naïve Bayes obtained an accuracy of 81.37% with a good recall 96.2% and precision 78.93%. The models of MultiLayer Perceptron, Random Forest, and Linear Kernel exhibit a moderate level of accuracy, precision, and recall. Random forest achieved a 93.51% accuracy with balanced recall of 95.71% and 96.47% precision. Random forest ranked the highest among the experimented machine learning models. RBF kernel obtained an overall higher accuracy of 91.16% with significantly greater precision, linear SVM achieved 89.58% accuracy, while poly kernel had the lowest accuracy of 69.32% but a substantial recall rate of 98.47%.

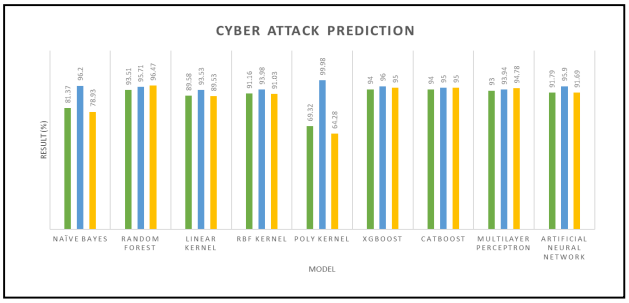


Fig. 5 Summary Graph

The models of Artificial Neural Network, XGBoost, CatBoost, RBF Kernel, and Poly Kernel show very good accuracy, recall, and precision near or over 95%. Notably, the Artificial Neural Network outperforms the other models in terms of precision, demonstrating how well it can reduce false positives when anticipating cyberattacks. All things considered, neural networks and ensemble tree-based models perform better than the other techniques in precisely detecting cyberattacks.

V. CONCLUSIONS & FUTURE SCOPE

The intentional concatenation of the training and testing sets highlights the dataset's utility for model training and evaluation. A comparative examination of numerous machine learning and deep learning algorithms was performed to determine their efficiency in terms of recall, precision, and accuracy metrics.

The support vector machine gave us an accuracy of 89%, whereas Random Forest and Naïve Bayes gave us scores of 93.51% and 81.37%, respectively.

The best accuracy was 94.79% using the XGBoost technique and 91.79% with deep learning algorithms. This paper summarizes the results for this dataset, showing that ensemble methods such as XGBoost and CatBoost give superior performance in detecting and preventing cyberattacks compared to other algorithms tested.

The project aims to develop an Application Programming Interface (API) for real-time threat detection and response. A centralized dashboard could aggregate insights, enabling more efficient monitoring and response. Custom tools or plugins could be developed to integrate with popular cybersecurity frameworks. The project could extend the scope of IDS to endpoint security and cloud-based environments.

REFERENCES

[1] Adebiyi, M. O., Arowolo, M. O., Archibong, G. I., Mshelia, M. D., & Adebiyi, A. A. (2021). An Sql Injection Detection Model Using Chi-Square with Classification Techniques. International Conference on Electrical, Computer, and Energy Technologies, ICECET 2021.

[2] Ahmed, M., & Uddin, M. N. (2020, December 19). Cyber Attack Detection Method Based on NLP and Ensemble Learning Approach. ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings.

[3] Alsahafi, R. (2019). SQL injection attacks: Detection and prevention techniques. International Journal of Scientific and Technology Research, 8(1), 182–185.

[4] Ashlam, A. A., Badii, A., & Stahl, F. (2022). Multi-Phase Algorithmic Framework to Prevent SQL Injection Attacks using Improved Machine learning and Deep learning to Enhance Database security in Real-time. Proceedings of the 2022 15th IEEE International Conference on Security of Information and Networks, SIN 2022.

[5] Chen, D., Yan, Q., Wu, C., & Zhao, J. (2021). SQL Injection Attack Detection and Prevention Techniques Using Deep Learning. Journal of Physics: Conference Series, 1757(1).

[6] Dānishgāh-i Shahīd Bihishtī, & Institute of Electrical and Electronics Engineers. (n.d.). 2020 11th International Conference on Information and Knowledge Technology (IKT) : December 22-23, 2020, Shahid Beheshti University - Tehran, Iran.

[7] De, S., & Sodhi, R. (2020, December 17). A simple cyber attack detection scheme for smart grid cyber security enhancement. 2020 21st National Power Systems Conference, NPSC 2020.

[8] Demilie, W. B., & Deriba, F. G. (2022). Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. Journal of Big Data, 9(1).

[9] Gogoi, B., Ahmed, T., & Dutta, A. (2021). Defending against SQL Injection Attacks in Web Applications using Machine Learning and Natural Language Processing. Proceedings of the 2021 IEEE 18th India Council International Conference, INDICON 2021.

[10] Hossain, F., Akter, M., & Uddin, M. N. (2021). Cyber Attack Detection Model (CADM) Based on Machine Learning Approach. International Conference on Robotics, Electrical and Signal Processing Techniques, 567–572.

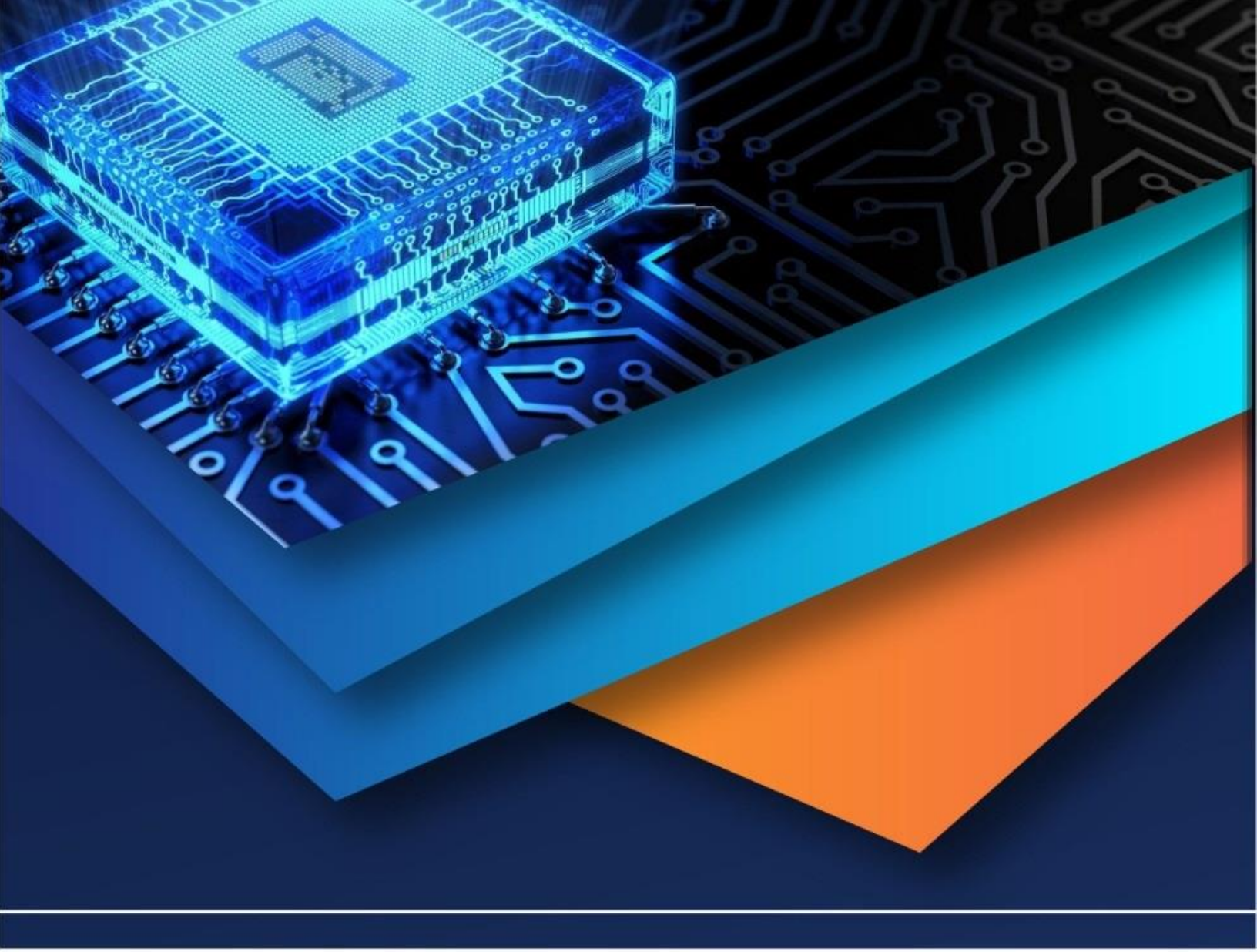
[11] Ross, K. (2018). SQL Injection Detection Using Machine Learning Techniques and Multiple Data Sources [San Jose State University].

[12] Sanjeev Agrawal Global Educational (SAGE) University, Institute of Electrical and Electronics Engineers. Madhya Pradesh Section, & Institute of Electrical and Electronics Engineers. (n.d.). Abstract Proceedings of International Conference on 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET) : 23rd-24th December 2022.

[13] Shahrivar, P. (2022). Detection of Vulnerability Scanning Attacks using Machine Learning Application Layer Intrusion Detection and Prevention by Combining Machine Learning and AppSensor Concepts.

[14] Su, Z. C., Hlaing, S., & Khaing, M. (n.d.). A Detection and Prevention Technique on SQL Injection Attacks.

[15] M. A. Helmiawan, E. Julian, Y. Cahyan and A. Saeppani, “Experimental evaluation of security monitoring and notification on network intrusion detection system for server security,” 2021 9th International Conference on Cyber and IT Service Management (CITSM), Bengkulu, Indonesia, 2021, pp. 1-6



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY