iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.53584

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 11 Issue VI Jun 2023- Available at www.ijraset.com

The Proposed Intrusion Detection System using Support Vector Machine by IOT Enabled WSN

Prof. Sumedh Dhengre[1], Anurag Khadtare[2], Pranav Kakani[3], Pratik Jagtap[4], Tushar Jambhulkar[5]
[1]*Professor, Dept of Computer Engineering, AISSMS COE, Pune, Maharashtra, India*
[2, 3, 4, 5]*Student, Dept of Computer Engineering, AISSMS COE, Pune, Maharashtra, India*

Abstract: Detecting intruders in computer networks is very important because it affects multiple communication and security domains finding network intruders can be difficult furthermore network intrusion detection remains a challenging undertaking due to the large amount of data required to train modern machine learning models to detect network intrusion risks recently many methods have been published for detecting network intruders however they face significant challenges as new threats continue to emerge that are undetectable by older systems this study evaluates different approaches to creating network intrusion detection systems the best features of the dataset are selected based on the correlations between features in addition we provide a complete functional and performance overview of an adaboost-based network attack detection solution based on these selected characteristics.

Keywords: AdaBoost, network intrusion, decision tree, SVM, MLP

I. INTRODUCTION

Intrusion detection systems (IDS) play a crucial role in network security by helping to identify and respond to potential threats. There are two main approaches that IDS use: anomaly detection and signature-based detection.

1) Anomaly Detection[4]: This approach focuses on identifying abnormal or suspicious behavior by comparing it to known patterns of normal activity. Anomaly detection systems analyze network traffic or host OS behavior using various characteristics, such as packet sizes, protocols, ports, or user behavior. If there is a significant deviation from the expected behavior the system triggers an alarm.

2) Anomaly detection is effective in detecting novel attacks or zero-day exploits, which do not have known signatures. However, it can also produce false positives if legitimate activity is mistakenly classified as anomalous. Fine-tuning anomaly detection systems and minimizing false positives can be a challenge.

3) Signature-Based Detection: This approach relies on a database of known attack signatures or patterns. The IDS compares incoming network traffic or system activity against the signatures in its database. If a match is found, indicating a known attack, an alert is generated. Signature-based detection is highly effective in identifying known threats and is particularly useful for detecting common attack methods. However, it has limitations when faced with new or modified attacks that do not match any existing signatures. Regularly updating the signature database is crucial to keeping up with emerging threats.

Based on their deployment, IDS can be categorized into two types:

a) *Network Intrusion Detection System (NIDS)[6]:* This type of IDS monitors network traffic by capturing packets from the network. The captured packets' headers are analyzed based on various factors, such as source and destination IP addresses, ports, protocols, and packet contents. NIDS can be deployed on servers, switches, gateways, or network backbones to monitor a wide range of network traffic.

b) *Host Intrusion Detection System (HIDS):* HIDS is installed on individual systems to monitor activities and events occurring on the host. It focuses on detecting unauthorized access attempts, modifications to system files, abnormal resource usage, unexpected process behavior, or other indicators of compromise. HIDS is especially useful for protecting critical systems or servers that may be targeted by attackers.

By combining NIDS and HIDS, organizations can create a layered defense strategy, where network-wide anomalies and known attack patterns are detected by NIDS, while HIDS provides granular monitoring and protection on individual hosts.

It's worth noting that while IDS can help in identifying potential threats, they are not foolproof and should be complemented with other security measures such as firewalls, antivirus software, secure configurations, and user awareness training to create a comprehensive security posture. Regenerate response.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

II. TYPES OF ATTACKS

Wireless sensor networks (WSNs)[21] are vulnerable to various types of attacks that impact power and CPU usage, as well as security. Compared to other networks, WSNs require specific solutions to address these attacks effectively. Here is a summary of each type of assault and its impact on WSNs:

1) *Eavesdropping:* Hackers can exploit the security limitations of WSNs, such as a hostile environment and unreliable communication, to intercept transmitted information between nodes. This attack increases the effects of radio fading and frequency transmission or dispersion.
2) *Collision:* Malicious nodes in WSNs can disrupt neighboring broadcasts by sending short noise packets, leading to potential network failures. These attacks do not follow the Intermediate Access Control Protocol and are difficult to trace due to wireless transmission characteristics.
3) *Unfairness:* Attackers exploit contract connection settings to block authorized users from accessing network resources and bypass submission deadlines. This can involve repeated collision attacks or the arbitrary exploitation of media access control layer priority schemes.
4) *Resource Depletion:* Collision attacks in WSNs are repeated until the nodes' energy is depleted, causing routing loops and energy exhaustion during packet transfers. This leads to a significant reduction in node energy.
5) *Traffic Analysis:* This attack targets sensitive data on sink or access point nodes by analyzing node communication patterns. It can result in the deletion of transactional processes on compromised nodes.
6) *Sybil:* Attackers simulate the existence of multiple sensor nodes by generating multiple node IDs from a single node. This leads to resource allocation problems and system failures, affecting load-balancing technologies and server protocols.
7) *Faking:* This attack manipulates routing data between nodes, causing routing loops, increased latency, network segmentation, and other disruptions.
8) *Session Espionage:* A man-in-the-middle attack that gives the attacker total control over an application account by taking over the session cookie. The attacker can access the user's account as long as the session token is retained on the user's device.
9) *Rejecting:* Attackers tamper with data authoring and forge additional steps in systems that lack proper monitoring and logging. This can result in inaccurate data logged in log files, leading to false or misleading information.

In summary, WSNs face various attacks that impact power, CPU usage, and security. Effective solutions must be developed to mitigate these attacks, considering the unique characteristics and constraints of WSNs. Regenerate response

III. ARCHITECTURE

WSNs (Wireless Sensor Networks) have a specific architecture called LR-WPAN (IEEE 802.15.4) that consists of five tiers, each providing different services:

1) The physical layer handles carrier generation, signal modulation, deflection, and data encryption.
2) The data link layer manages media access control, data multiplexing, framing, and point-to-point communication.
3) The network layer controls address distribution and packet forwarding.
4) The transport layer ensures secure packet transmission.
5) The application layer handles data representation between sensor nodes and between nodes and end users.

For WSNs, connecting to the internet is challenging due to their low power nature. The 6LoWPAN architecture addresses this by adding an adaptation layer between the network and data link layers. It allows the use of various address spaces and facilitates the compatibility of IPv6 packets over existing IPv4 networks. The 6LoWPAN edge router handles packet forwarding at the network layer, reducing energy usage and eliminating the need to maintain application layer state.

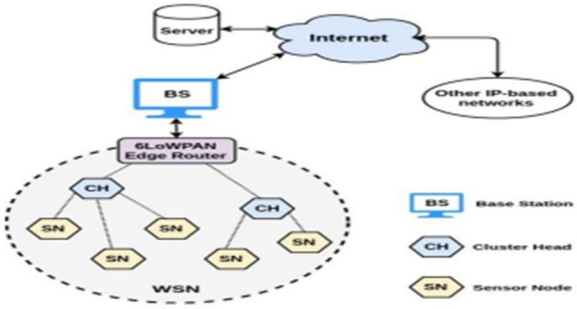


Fig 3.1 LoWPAN

However, connecting low-powered devices to the internet raises security concerns. The integration of WSNs with the internet has led to the development of new applications such as the Internet of Everything (IoE) and the Internet of Things (IoT), which are not covered in detail here.

IV. METHODOLOGY

A. *SVM*

Support Vector Machine (SVM) is a popular algorithm used in supervised learning for classification and regression problems. Its goal is to determine an optimal decision boundary or hyperplane that can separate data points into different classes.

The key components of SVM include:

1) *Support Vectors:* These are the data points closest to the hyperplane and have the most influence on its orientation and position. They are used to maximize the margin of the classifier and play a crucial role in building the SVM model.
2) *Negative Hyperplane:* This refers to the plane to which most of the negative data points belong.
3) *Positive Hyperplane:* This refers to the plane to which most of the positive data points belong.
4) *Margin:* The margin is the distance between the hyperplane and the closest observations (support vectors). In SVM, a larger margin is considered better. There are two types of margins: hard margin and soft margin.

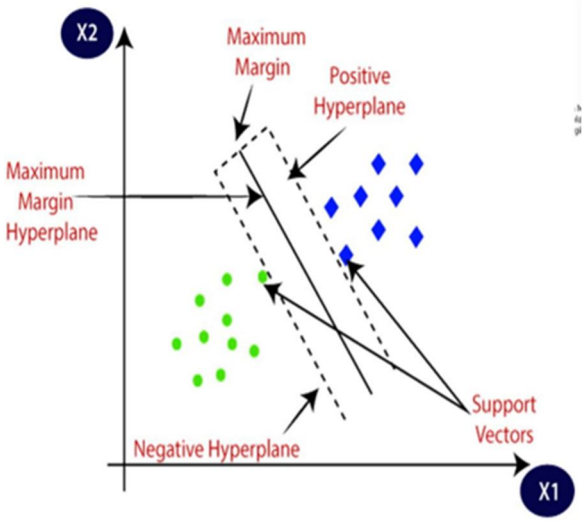


Fig 4.1 SVM Architecture

There are two main types of SVM:

a) *Linear SVM[9]:* In this type, the data points can be separated by drawing a plane or hyperplane between the two classes. It works well when the data is linearly separable.
b) *Non-linear SVM:* In cases where the data points cannot be separated by a simple linear hyperplane, non-linear SVM is used. It employs techniques such as the kernel trick to transform the data into a higher-dimensional space where a linear decision boundary can be applied.

Mathematically, SVM involves optimizing an objective function with constraints. The goal is to find the hyperplane that maximizes the margin while satisfying the classification constraints. Soft margin SVM allows for some misclassifications to handle non-linear or almost linearly separable datasets.

Kernels play a crucial role in SVM, allowing it to work with non-linear datasets. Different types of kernels, such as the polynomial kernel, sigmoid kernel, and radial basis function (RBF) kernel, can be used to transform the data into higher-dimensional feature spaces where linear separation becomes possible.

Overall, SVM is a powerful algorithm for classification and regression tasks, capable of handling both linearly separable and non-linear datasets through the use of hyperplanes and kernel functions.

B. Random Forest

The Random Forest[17] algorithm is a popular machine learning technique used for intrusion detection. It is an ensemble learning method that combines multiple decision trees to make predictions. Here's a brief explanation of how the Random Forest algorithm works for intrusion detection:

1) *Ensemble of Decision Trees:* Random Forest consists of a collection of decision trees, where each tree is trained on a random subset of the training data. Each tree independently makes a prediction, and the final prediction is determined by a majority vote or averaging over the predictions of all the trees.
2) *Random Feature Selection [18]:* When constructing each decision tree, a random subset of features (columns) is selected as candidate features at each split point. This helps to reduce the correlation among the trees and promotes diversity in the ensemble.
3) *Bagging:* Random Forest uses a technique called bagging (bootstrap aggregating), where each tree is trained on a bootstrap sample of the training data. A bootstrap sample is created by randomly selecting data instances from the original dataset with replacement. This sampling technique introduces randomness and reduces overfitting.
4) *Voting or Averaging:* Once all the decision trees are trained, they collectively make predictions on new instances. In the case of classification for intrusion detection, the final prediction is determined by majority voting, where the class that receives the most votes from the individual trees is chosen. For regression problems, the predictions from all the trees are averaged to obtain the final prediction.
5) *Importance of Features:* Random Forest can also provide a measure of feature importance. By evaluating how much each feature contributes to the accuracy of the model, it helps in identifying the most relevant features for intrusion detection.

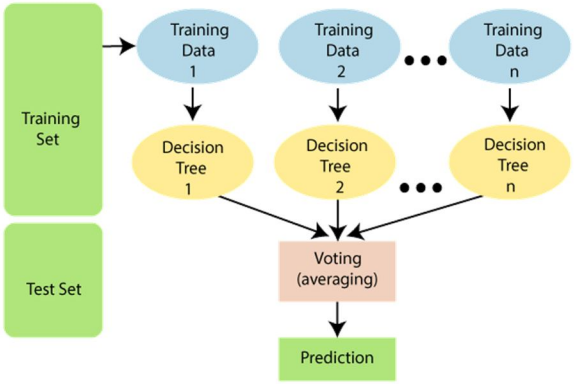


Fig 4.2 random Forest Architecture

Random Forests are known for their ability to handle large datasets, high-dimensional feature spaces, and noisy data. They are robust against overfitting and can handle both categorical and continuous features. Random Forests have shown good performance in detecting anomalies and classifying network traffic for intrusion detection tasks.

In summary, Random Forest is an ensemble learning algorithm that combines multiple decision trees trained on random subsets of data and features. It offers accurate and robust intrusion detection by leveraging the collective predictions of multiple trees.

V. WORKING

A. SVM

Support Vector Machine (SVM)[3-9] is a machine learning algorithm that can be used for intrusion detection in network security. Here is a simplified explanation of how SVM works in the context of intrusion detection:

1) *Data Preparation:* The first step is to gather a labeled dataset for training the SVM model. This dataset consists of network traffic data, where each instance is labeled as either normal or intrusive behavior.
2) *Feature Extraction:* From the network traffic data, relevant features are extracted. These features can include source and destination IP addresses, port numbers, packet sizes, protocol types, etc. The goal is to represent each network connection or packet as a set of numerical features.
3) *Training the SVM Model:* The extracted features and corresponding labels are used to train the SVM model. The SVM algorithm aims to find an optimal hyperplane that separates the normal instances from the intrusive instances in the feature space. The hyperplane is chosen in such a way that the margin between the closest data points from each class is maximized.

4) *Mapping to High-Dimensional Space:* In some cases, the original feature space may not be linearly separable. SVM uses a technique called the kernel trick to map the data into a higher-dimensional space where separation becomes possible. Common kernel functions used in SVM for intrusion detection include the Radial Basis Function (RBF) kernel.
5) *Classification:* Once the SVM model is trained, it can be used to classify new, unseen instances of network traffic as either normal or intrusive. The model examines the feature representation of each instance and assigns it to one of the two classes based on its position relative to the learned hyperplane.
6) *Detection and Decision:* The output of the SVM model can be interpreted as a decision function or probability scores, depending on the specific SVM variant used. A threshold can be set to determine the threshold for classifying an instance as intrusive. If the output exceeds the threshold, the instance is flagged as potentially intrusive.
7) *Evaluation and Refinement:* The performance of the SVM model is evaluated using evaluation metrics such as accuracy, precision, recall, and F1 score. If necessary, the model can be refined by adjusting parameters or using techniques like cross-validation.

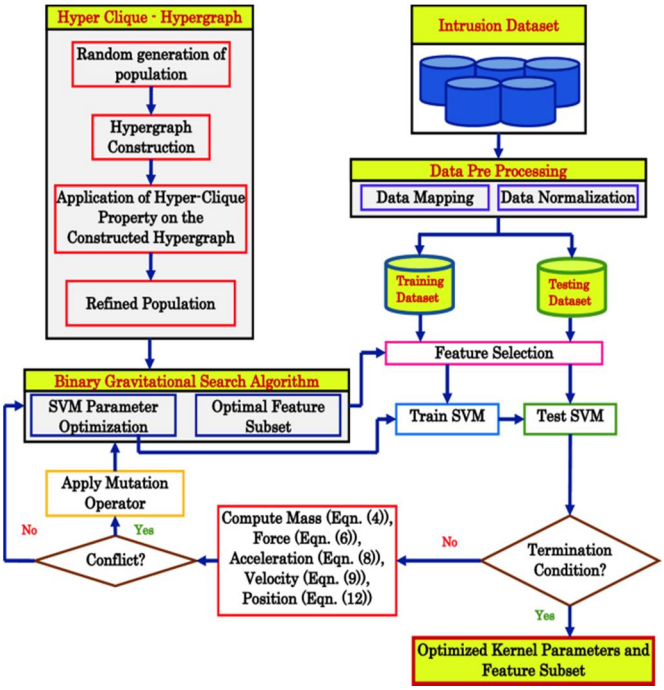


Fig 5.1 Working of SVM

It's important to note that SVM is just one of many algorithms used in intrusion detection. The choice of algorithm depends on the specific requirements, the nature of the data, and the available resources.

B. Random Forest

The Random Forest algorithm is a popular machine learning technique used for intrusion detection. It is an ensemble learning method that combines multiple decision trees to make predictions. Here's a brief explanation of how the Random Forest algorithm works for intrusion detection:

1) *Ensemble of Decision Trees[14]:* Random Forest consists of a collection of decision trees, where each tree is trained on a random subset of the training data. Each tree independently makes a prediction, and the final prediction is determined by a majority vote or averaging over the predictions of all the trees.
2) *Random Feature Selection:* When constructing each decision tree, a random subset of features (columns) is selected as candidate features at each split point. This helps to reduce the correlation among the trees and promotes diversity in the ensemble.
3) *Bagging:* Random Forest uses a technique called bagging (bootstrap aggregating), where each tree is trained on a bootstrap sample of the training data. A bootstrap sample is created by randomly selecting data instances from the original dataset with replacement. This sampling technique introduces randomness and reduces overfitting.

4) *Voting or Averaging:* Once all the decision trees are trained, they collectively make predictions on new instances. In the case of classification for intrusion detection, the final prediction is determined by majority voting, where the class that receives the most votes from the individual trees is chosen. For regression problems, the predictions from all the trees are averaged to obtain the final prediction[10].
5) *Importance of Features:* Random Forest can also provide a measure of feature importance. By evaluating how much each feature contributes to the accuracy of the model, it helps in identifying the most relevant features for intrusion detection.

Random Forests are known for their ability to handle large datasets, high-dimensional feature spaces, and noisy data. They are robust against overfitting and can handle both categorical and continuous features. Random Forests have shown good performance in detecting anomalies and classifying network traffic for intrusion detection tasks.
In summary, Random Forest is an ensemble learning algorithm that combines multiple decision trees trained on random subsets of data and features. It offers accurate and robust intrusion detection by leveraging the collective predictions of multiple trees.
Building and Training SVM Mode[10]l: The code defines the function SVM() to build an SVM model with a linear kernel. It utilizes the training data to train the SVM model. The steps involved in training the SVM model are as follows:
Creating an instance of the SVM classifier using SVC class from scikit-learn.Fitting the classifier to the training data using the fit() method. This step involves finding the optimal hyperplane that maximizes the margin between the two classes (intrusion and normal).
Making Predictions: After training the SVM model, the code uses the trained model to make predictions on the test data. The steps involved in making predictions are as follows:
Using the predict() method of the SVM classifier to predict the labels of the test data based on the learned hyperplane. Storing the predicted labels in a variable.
Evaluating the Model: The code calculates evaluation metrics such as accuracy, classification report, and confusion matrix to assess the performance of the SVM model. These metrics provide insights into how well the model has classified the instances into the correct classes (intrusion or normal).
Saving the Model: Once the SVM model is trained and evaluated, the code saves the model as a joblib file (svm_new.joblib) using the joblib.dump() function. This allows the model to be loaded and used later without retraining.

VI. ACCURACY

	Precision	Recall	F1-score	Support
Accuracy			0.92	485,268
Macro average	0.89	0.87	0.88	485,268
Weighted average	0.92	0.92	0.92	485,268

Classification Report	Precision	Recall	F1-score	Support
Neptune	0.93	0.81	0.86	107,201
normal	0.79	0.79	0.79	97,277
smurf	0.96	1.00	0.98	280,790

VII. CONCLUSION

The proposed work aims to identify and analyse the threats and issues by making use of algorithms like SVM. The increasing frequency of intrusion crimes necessitates the development of an optimal intrusion detection system (IDS) that surpasses traditional clustering algorithms. This study successfully addressed this need by developing an IDS using the Support Vector Machine (SVM) algorithm, demonstrating its accuracy and efficiency in detecting intrusions.

The research also delved into the broader context of security, emphasizing the importance of IDS in the realm of machine learning (ML) and deep learning (DL) techniques applied to Wireless Sensor Networks (WSNs). This is highly relevant as the proliferation of interconnected devices in the Internet of Things (IoT) has introduced new vulnerabilities and heightened the need for robust intrusion detection systems.

By considering well-known IDS datasets, the study further enhanced its practicality and provided valuable resources for training and evaluating the proposed SVM-based IDS. These datasets serve as benchmarks for researchers and practitioners, allowing them to develop and compare different methodologies and techniques in intrusion detection.

Importantly, this research not only contributes to the field of intrusion detection but also highlights the significance of interdisciplinary approaches. By incorporating knowledge from ML, DL, WSNs, and IoT, the study underscores the importance of integrating various domains to develop comprehensive and effective security solutions.

Furthermore, the project's exploration of future research directions serves as a valuable resource for inspiring researchers to explore unexplored areas and drive advancements in intrusion detection technology. This includes potential avenues such as the integration of anomaly detection techniques, the incorporation of real-time data streams, or the application of explainable AI methodologies to enhance transparency and interpretability in IDS.

In summary, this research makes a significant contribution to the field of intrusion detection by introducing an effective IDS based on SVM and providing insights into related areas such as ML, DL, WSNs, and IoT. By incorporating relatable points and outlining future research directions, this project serves as a valuable resource, inspiring researchers to explore new realms and drive further advancements in intrusion detection technology.

VIII. FUTURE SCOPE OF THE PROJECT

The future prospects of intrusion detection systems (IDS) are promising, offering several areas for potential development and improvement.

Firstly, the integration of advanced machine learning and deep learning techniques can significantly enhance the detection capabilities of IDS. By utilizing algorithms such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), IDS can achieve more precise and efficient identification of network intrusions, including previously unseen or zero-day attacks.

Secondly, the incorporation of big data analytics and data mining techniques can empower IDS to handle the vast amounts of data generated by modern networks. This capability can lead to more effective detection of complex attack patterns and the identification of sophisticated intrusion attempts.

Furthermore, expanding the application of IDS to new domains and industries holds great potential. For example, deploying IDS in emerging technologies like Internet of Things (IoT) networks, cloud computing environments, and industrial control systems can provide enhanced security and protect critical infrastructures.

Moreover, the development of collaborative and distributed IDS frameworks can enable the sharing of threat intelligence and facilitate collaborative analysis of network data. This approach fosters a more robust and comprehensive defences against cyber threats.

In summary, the future scope of intrusion detection systems lies in leveraging advanced techniques, effectively handling big data, exploring new domains, and promoting collaboration to continually enhance the effectiveness and efficiency of intrusion detection. These advancements ultimately strengthen the security of networked systems.

IX. APPLICATIONS

The future scope of an Intrusion Detection System (IDS) using Support Vector Machine (SVM) in an IoT-enabled Wireless Sensor Network (WSN) is promising and holds several potential developments and improvements:

1) *Enhanced Detection Accuracy:* The future of IDS using SVM in IoT-enabled WSN lies in improving the detection accuracy of intrusion attempts. This can be achieved by refining the SVM algorithm, optimizing feature selection, and incorporating advanced techniques such as ensemble learning or deep learning architectures specifically designed for WSNs.

2) *Real-time and resource-efficient Detection:* As IoT-enabled WSNs operate in resource-constrained environments, the future IDS should focus on developing lightweight and energy-efficient algorithms. This includes exploring techniques like online learning, adaptive model updating, or distributed detection to enable real-time intrusion detection while minimizing resource consumption.
3) *Anomaly Detection for IoT-specific Attacks:* With the proliferation of IoT devices, new attack vectors and IoT-specific vulnerabilities emerge. The future IDS should be capable of detecting anomalous behavior and identifying attacks that specifically target IoT devices, such as IoT botnets, device spoofing, or firmware tampering.
4) *Integration with edge Computing:* As edge computing becomes more prevalent in IoT deployments, IDS using SVM in IoT-enabled WSN can benefit from localized detection and decision-making. By performing intrusion detection at the network edge, near the data source, IDS can reduce latency, enhance privacy, and improve overall network security.
5) *Adaptability to Dynamic WSN Environments:* IoT-enabled WSNs are often subject to dynamic network conditions, topology changes, or mobility of nodes. The future IDS should be designed to adapt to these dynamic environments by incorporating techniques like transfer learning, online training, or context-aware detection to maintain high detection accuracy even in evolving WSN scenarios.
6) *Integration with threat intelligence and collaborative defense:* IDS in IoT-enabled WSNs can leverage threat intelligence feeds and collaborate with other IDS systems or security platforms to enhance its detection capabilities. By sharing threat information, updating attack signatures, or participating in collective defense efforts, the IDS can stay updated on emerging threats and improve its overall effectiveness.

In summary, the future scope of IDS using SVM in IoT-enabled WSN involves advancing detection accuracy, optimizing resource efficiency, addressing IoT-specific attacks, integrating with edge computing, adapting to dynamic environments, leveraging threat intelligence, and ensuring privacy in IoT deployments. These advancements will contribute to more robust and secure intrusion detection in IoT-enabled WSN environments.

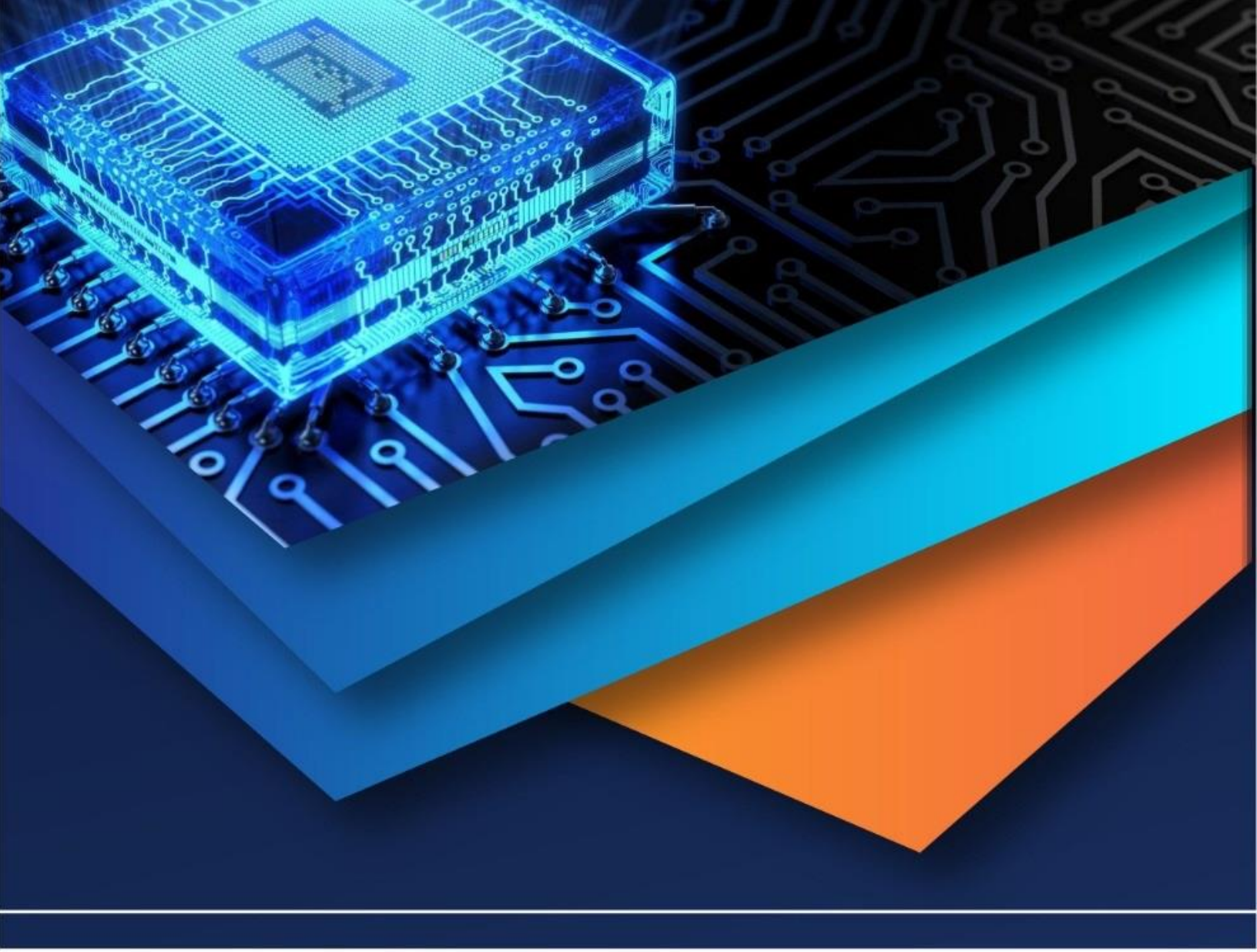
X. RESULTS

In this section, we present the results of our study on an Intrusion Detection System (IDS) using Support Vector Machine (SVM) within an IoT-enabled Wireless Sensor Network (WSN). Our evaluation aims to assess the performance and effectiveness of the proposed IDS in detecting and classifying various types of intrusions in real-time. We begin by providing an overview of the performance evaluation metrics used in our study. Subsequently, we recap the experimental setup and methodology employed. Finally, we present the performance results of our IDS, comparing its performance with other existing methods and providing insights into its robustness, generalizability, and computational efficiency. Through these findings, we gain a comprehensive understanding of the IDS's performance and its potential as a reliable security mechanism in IoT-enabled WSN environments.

REFERENCES

[1] Jayshree Jha and Leena Ragha(2013).Intrusion Detection System using Suppoort Vector Machine.
[2] Nitish A ,Hanumanthappa J , P. Deepa Shenoy ,K.R. Venugopal(2019).Aspects of Machine Learning based Intrusion Detection Systems in Wireless Sensor Networks:A Review.
[3] Zakiyabanu S. Malek, Bhushan Trivedi, Axita Shah(2020).User behavior Pattern -Signature based Intrusion Detection.
[4] Anand Sukumar J V, Pranav I, Neetish MM, Jayasree Narayanan (2018).Network Intrusion Detection Using Improved Genetic k-means Algorithm
[5] Dr. Manish Kumar and Ashish Kumar Singh(2020).Distributed Intrusion Detection System using Blockchain and Cloud Computing Infrastructure.
[6] WEN-TAOLIU(2008).RESEARCH ON INTRUSION DETECTION RULES BASED ON XML IN DISTRIBUTED IDS
[7] Zhan Xin,Wang Xiaodong and Yuan Huabing(2019).Research on Block Chain Network Intrusion Detection System
[8] Shan Suthaharan(2012).An Iterative Ellipsoid-Based Anomaly Detection Technique for Intrusion D.
[9] "Intrusion Detection System using Support Vector Machine and Random Forest with Feature Selection" by Ramandeep Kaur and Naveen Dhillon. Published in the International Journal of Computer Applications, Volume 180, Issue 4, 2018.
[10] "Anomaly Intrusion Detection using One-Class Support Vector Machine with Deep Learning" by Shrikant Ravikar, Virendra Singh, and R. K. Agrawal. Published in the Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS), 2018.
[11] "Intrusion Detection in IoT Networks Using Support Vector Machines" by Meiling Qi, Xiaoxue Li, and Jiguo Yu. Published in the Proceedings of the International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2017.
[12] "Intrusion Detection System Using Support Vector Machine and Principal Component Analysis" by Vipul Dalal, Shishir K. Shandilya, , 2014.
[13] Alaba, F. A., et al. (2017). "Intrusion detection systems in wireless sensor networks: A review." Journal of Network and Computer Applications, 88, 10-25.
[14] García-Teodoro, P., et al. (2009). "Anomaly-based network intrusion detection: Techniques, systems and challenges." Computers & Security, 28(1-2), 18-28.
[15] Subramanian, T., et al. (2014). "A comprehensive study on classification of network attacks in cloud environment using support vector machines." Procedia Engineering, 97, 2252-2261.
[16] KDD Cup 1999 Data. Retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[17] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[18] Hastie, T., et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
[19] Chen, M., et al. (2016). "Intrusion detection techniques for IoT: A systematic review and meta-analysis." Journal of Network and Computer Applications, 75, 198-207.
[20] Raza, S., et al. (2013). "Wireless sensor networks for intrusion detection: Packet traffic modeling." Ad Hoc Networks, 11(2), 879-894.
[21] Patel, V. M., et al. (2016). "Intrusion detection techniques in wireless sensor networks: A review." Computer Communications, 89, 1-19.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY