Babylonian Journal of Artificial Intelligence
Vol. 2024, **pp**. 158–167
DOI: https://doi.org/10.58496/BJAI/2024/017 ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BJAI

Research Article

Enhancing IoT Security with AI-Driven Hybrid Machine Learning and Neural Network-Based Intrusion Detection System

Thaker Nay[1],*,

[1] *Luleå University of Technology, Sweden.*

ARTICLE INFO

Article History

Received 12 Aug 2024
Revised: 11 Sep 2024
Accepted 10 Nov 2024
Published 05 Dec 2024

Keywords

KDD

DS2OS

IoT Botnet datasets

SVM

ABSTRACT

The increasing occurrence of cyberattacks specifically aimed at critical infrastructure has led to the adoption of network intrusion detection techniques for the Internet of Things (IoT). Securing IoT networks is difficult because of the growing number of connected devices and the advanced methods used by attackers. This study investigates the application of machine learning and neural networks in the prevention of prevalent online fraud and assesses their efficacy. The text discusses important ideas related to email filtering, machine learning, artificial neural networks, and network intrusion techniques. The study discusses the difficulties related to e-fraud detection and suggests methods to improve detection systems. Furthermore, it offers a thorough examination of IoT intrusion detection, emphasizing the risks, weaknesses, assaults, and methods of detection. Securing the billions of autonomous nodes in the Internet of Things (IoT), each with distinct characteristics, poses a significant challenge. Conventional techniques like as encryption, access control, and authentication are inadequate when used individually. Thus, this work utilizes deep learning techniques to detect widespread IoT vulnerabilities, such as Distributed Denial of Service (DDoS) assaults. The models are evaluated using different datasets, including NSL-KDD, DS2OS, and IoT Botnet. The evaluation is based on measures such as accuracy, precision, recall, and F1-score. The deep machine learning intrusion detection system has a high accuracy rate of 96.38%, which shows its efficiency in recognizing risks related to the Internet of Things (IoT) Where the data was trained by 80% and the data was tested by 20.

1. INTRODUCTION

The rise of cyberattacks on wireless sensor networks (WSN) necessitates advanced detection and categorization methods. Artificial Intelligence (AI) plays a critical role in this domain, enabling the use of hybrid feature reduction strategies that integrate deep learning and machine learning techniques [1]. One such strategy combines K-means clustering with entropy-based mutual information feature ranking, effectively reducing the high-dimensional feature space by identifying and prioritizing the most relevant features [2,4]. Subsequently, a feed-forward deep neural network (DNN) is employed to train the algorithm to accurately classify network traffic. Early detection and high-performance learning systems are vital for successfully identifying and mitigating cyberattacks in WSN environments [5,10].

Despite the potential of AI, WSN faces numerous cybersecurity challenges. Traditional security measures often fall short, highlighting the need for advanced machine learning (ML) approaches to detect and respond to threats [11]. Effective cybersecurity solutions must protect data, servers, networks, computers, and information systems from WSN-related attacks [12,16]. Implementing robust cybersecurity policies ensures the confidentiality, integrity, and availability of sensitive data, guarding against unauthorized access by cybercriminals [17].

Cybersecurity encompasses the protection of computer and mobile networks, software, servers, and electronic devices from malware and viruses. The escalating global threat of cybercrime mandates stringent cybersecurity measures [18]. Machine learning, a subset of AI, is employed in prediction systems and zero-day attack detection, utilizing various methods such as supervised, semi-supervised, unsupervised, and reinforcement learning to combat cyberattacks [19]. Deep learning, involving multi-stage training on diverse datasets, further enhances cybersecurity efforts.

*Corresponding author. Email: thaker.nayl@uoanbar.edu.iq

Intrusion Detection Systems (IDS) are crucial for safeguarding systems against covert vulnerabilities. Support Vector Machine (SVM) has gained traction in IDS design due to its notable success. However, optimizing SVM parameters poses challenges. Particle Swarm Optimization (PSO), a global search optimization technique, is used to fine-tune SVM parameters and select features for IDS [20,21].
Integrating AI, machine learning, and deep learning techniques into WSN environments and cybersecurity applications significantly enhances cyberattack detection and prevention, ensuring the security of both data and systems [22].

2. LITERATURE REVIEW

In this paragraph. Previous studies have explored various methods for achieving this.
Sharma, B., et al. [23] conducted a cyberattack evaluation using deep learning techniques integrated with explainable artificial intelligence (XAI) for IDS in IoT networks. Their approach provided transparency in decision-making, aiding experts in understanding the model's behavior. The method achieved high accuracy, but its complexity and computational demands were notable drawbacks.
Sun, Z., et al. [24] investigated security issues within IoMT. They enhanced IDS using machine learning algorithms, achieving improved detection rates. However, the study highlighted the need for real-time processing capabilities and robustness against evolving threats.
Ali, S., et al. [25] focused on protecting WSNs from attacks using blockchain and federated learning-based IDS for edge-enabled industrial IoT networks. Their hybrid approach improved security and reduced latency but faced scalability challenges.
Emil Selvan and colleagues [26] utilized a hybrid optimization and integrated deep Q network to detect and mitigate network intrusions. This method showed promising results in reducing false positives and enhancing detection accuracy, though it required significant computational resources.
Alwahedi, F., et al. [27] explored the challenges of IoT security through the lens of generative artificial intelligence and large language models. Their review highlighted current research trends and future prospects, emphasizing the potential of AI to address complex security issues but also noting the risks of over-reliance on automated systems.
Liao, H., et al. [28] provided an overview of deep learning methods for IoT intrusion detection. They discussed various architectures like CNNs and RNNs, which achieved high detection rates but required extensive training data and suffered from interpretability issues.
Nanjappan, M. proposed DeepLG SecNet, a combination of secure networks with deep LSTM and GRU, enhancing intrusion detection in IoT environments [29]. The model showed significant improvements in accuracy and detection speed, though it required optimization for large-scale deployment.
Kumar, G. S. C., et al. [30] and Binbusayyis, A., et al. [31] investigated deep residual convolutional neural networks for IDS, demonstrating high efficiency in detecting anomalies. Their methods, while effective, required substantial computational power and data preprocessing.
Gaganjot et al. [32] proposed combining hybrid VGG19 and 2D-CNN architectures to detect intrusions in FOG-cloud environments. Their approach improved detection rates and robustness, but faced challenges in balancing computational overhead with real-time processing needs.
Manocchio, L. D., et al. [33] developed a transformer-based architecture for flow-based network intrusion detection systems, termed Flow Transformer. This method enhanced detection accuracy and scalability but required further testing in diverse IoT environments.

3. SUPPORT VECTOR MACHINE (SVM)

In the 1990s, Vapnik and his associates developed the Support Vector Machines (SVM) hypothesis. Neural networks provided the inspiration for SVM, or more accurately, SVM is a mathematical extension of neural networks. Both linear and nonlinear input may be correctly classified using Support Vector Machines (SVM). As seen in Figure 1, the SVM algorithm creates a hyperplane in a higher-dimensional space by mapping the initial training data onto it. The foundation of SVM, a mathematical learning method, is an optimal hyper-plane [17].

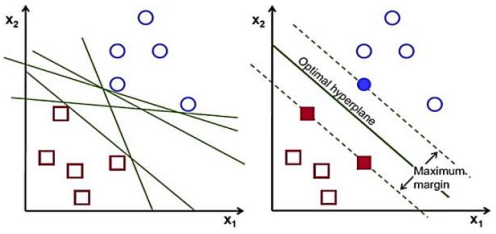

Fig. 1. performs class_of SVM

The goal of the support vector machine method is to locate a hyperplane in N-dimensional space (where N is the number of features) that effectively splits the data points into different classes [17].

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle) \quad (1)$$

$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k \quad (2)$$

$$\frac{\delta}{\delta w_k} (1 y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \quad (3)$$

Several of these hyperplanes can be employed to divide the two data sets. Finding the plane with the largest margin, or separation between points in different classes, is the objective. Accordingly, by opti]mizing the margin distance between two sample classes, we might potentially enhance the classification of fresh data points and deliver more trustworthy findings.

3.1 Hyperplanes and support vectors

When classifying data points, mathematical structures called hyperplanes serve as decision boundaries. Various categories may be applied to points on each side of the hyperplane. The hyperplane's size is also dependent on the feature count. When two input characteristics are provided, the hyperplane assumes the shape of a straight line. When three input characteristics are present, the hyperplane is replaced by a two-dimensional plane. The visibility decreases as the number of components increases above three (Figure 2).

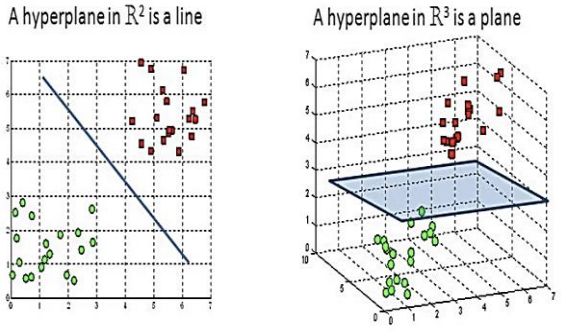

Fig. 2. Hyperplane of 2D and 3D

Support vectors are specific data points that are located close to the hyperplane and have a substantial impact on the orientation and positioning of the hyperplane. With these support vectors, we may increase the classifier's margin to its maximum potential. If the support vectors are removed, the location of the hyperplane will change. The following rules are fundamental for developing our Support Vector Machine (SVM), as shown in Figure 3.

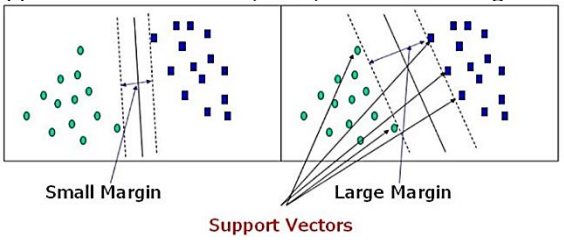

Fig. 3. Comparison of Small and Large Margins in Support Vector Machines

4. METHODOLOGY

The proposed approach propose a dual-layered, hybrid intrusion detection system (IDS). Rules-based detection in the first layer is followed by a supervised learning intrusion detection system (IDS) in the second layer. Figure 1 proposes for the Hybrid IDS approach with machine learning.

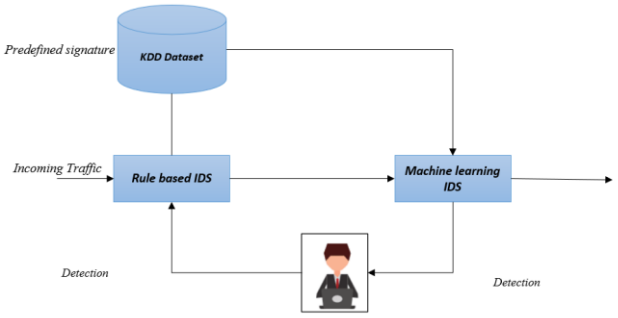

Fig. 4. Framework for Intrusion Detection System Using Rule-Based and Machine Learning Approaches

A local store of harmful signatures, such as the DShield Block List, ATLAS from Arbor Networks, and DGA list, will be available to Rule Based Intrusion Detection Systems. Other kinds of signatures, including spam, trash, phishing, and so on, may be found in databases like lot.ndb, push.Ndb, foxhole all.Cdb, junk.DB, and so on. Every packet originating from the traffic is divided into content and IP headers. Consider the IP headers as signatures, then search the database for malicious IPs that match them. After that, the text may be examined further to produce further signatures, such as a section on spam emails and other signature kinds that have already been covered. The user can be alerted to the intrusive party and take appropriate action after discovering a malicious signature. The subsequent layer receives the legitimate traffic. Because gradient descent is so much quicker than analytical approaches, it is used instead. The temporal complexity of analytical procedures is O(n3), making them unsuitable for use in real-world scenarios . Whether or not the input crosses a certain threshold determines the projected outcome in an SVM model. The input is regarded as abnormal if it exceeds the threshold and vice versa.. The sigmoid function defines the output and is defined as [-1 0 1]. A threshold value is chosen based on a variety of restrictions. Assume that 0.5 is the threshold. Thus, if y equal one , and y equal zero as well as X0 (X) more than or equal 0.5 ,y is equal to one, and y is also equal to zero, whereas X0 (X) is greater than or equal to 0.5. In this way, we can forecast the result based on the threshold. Set the threshold value at around 0.3 to reduce the frequency of false positives, but keep in mind that it could miss certain assaults. The false positive rate will be significant but the majority of assaults will be covered if the threshold value is set at around 0.7. As mentioned before, the method checks for intrusions by first collecting the packets and then executing a hybrid algorithm. As a result, the algorithm may be split into the subsequent stages:

4.1 Packet Capturing

Capturing packets is the initial step in the intrusion detection process. A Java version of Pcap called jNetPcap may be used for this. After reading every network interface, it chooses one and reads every packet that is coming in over that interface. After then, the packet may be examined to verify its content, protocol, source and destination addresses, and use. The PROMISCUOUS MODE will be used to capture packets. This implies that every packetincluding ones meant for machines other than the host will be caught. It will be possible to handle each packet using the functionalities provided by a PcapPacketHandler. After that, the packets may be utilized to obtain the content and packet header. The PcapPacketHandler will be used to build up a loop that will determine how many packets need to be collected. Figure 2 Packet filtering module.

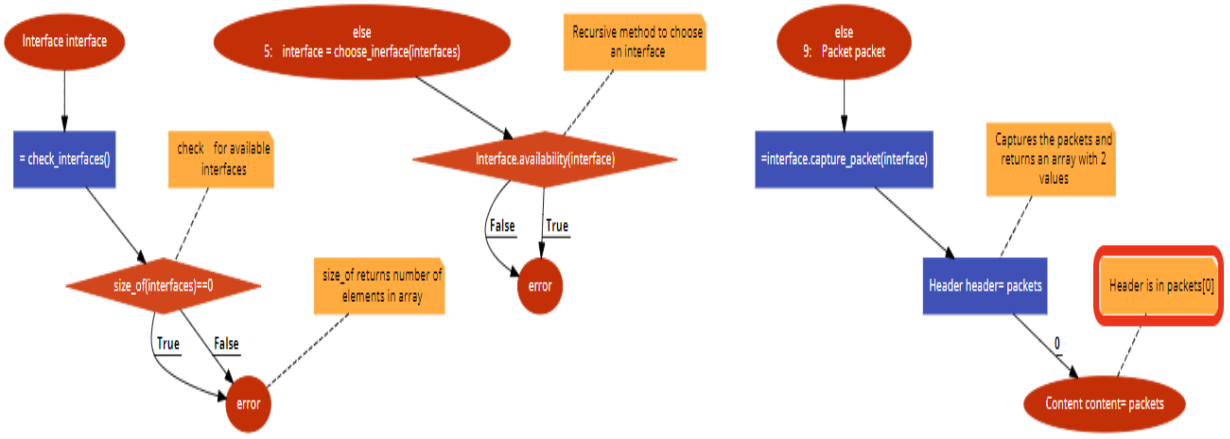

Fig. 5. Flowchart of Interface Selection and Packet Capture Process in Network Systems

4.2 Signature-Based Intrusion Detection

Predefined signatures will be examined in the packet headers. An alert informing the user of the sort of intrusion will be sent if the malicious signatures and the packet header match. For every sort of signature, the packet headers may be compared and the signatures can be defined independently. Figure 3 detection of signature-based intrusions.

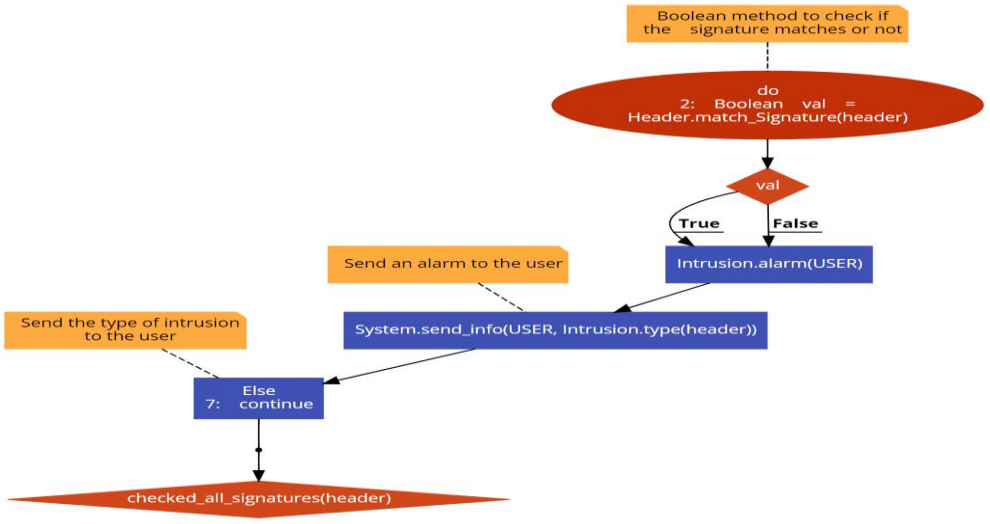

Fig. 6. Detection of signature-based intrusions.

4.3 Anomaly Based Intrusion Detection

The suggested machine learning model will be used by this method to operate. A packet data set will be used to train the model. Three sets of training data may be created, and the cost functions from each set can be used to calculate the training error. Figure 6 Anomaly Based Intrusion Detection.

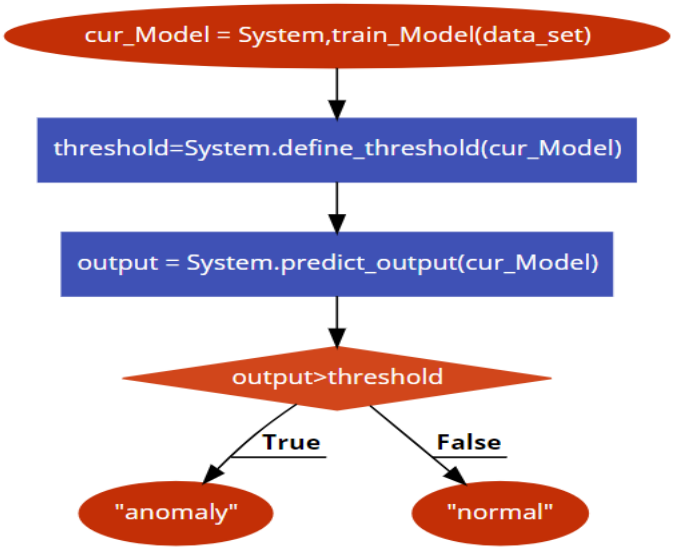

Fig. 6. Anomaly Based Intrusion Detection.

4.4 IOT Network Model

The intrusion detection system (IDS), which must be customized for this architecture, is subject to a number of limitations by the Internet of Things (IoT). Among these limitations are:

1. Sensor nodes' limited battery life limits the amount of energy-saving calculations that may be done.
2. High-end computing skills are not viable due to the limited computational resources available on sensor nodes.
3. When nodes are not in use, they go inactive, which makes it impossible to deploy an IDS that is always active.

This network has to be built with these limitations in mind in order to create an IDS for it. It need to adjust to the state of each node, carry out calculations with the restricted processing power, and take into account each node's finite battery life.

For the IDS to monitor local node traffic and that of nearby nodes and provide network insights, detection agents must be deployed. Reporting any identified disruption in network traffic to the base station is necessary so that it can notify the user of the alarm. The base station can receive notifications more easily thanks to the underlying network design.
Nodes should provide information to the IDS, such as local alert repositories and information on suspect nodes. Furthermore, pre-generated neighbor knowledge can be applied. Information about the warning's creation time, categorization, and source should all be included in the internal alert database.

5 EXPARMANTAL RESULT

The effectiveness of the proposed approach has been verified through extensive experiments using real-world IoT datasets. Evaluation metrics such as accuracy, false positive rates and detection rates are used to evaluate the performance of the IoT detection system as shown below:

5.1 Analysis of NSL-KDD dataset

Every connection record in the NSL-KDD dataset comprises 42 properties, including a class label that lists the different sorts of attacks. Mahbod Tavallaee et al. classified the attack types into three attack classes in A Detailed study of the KDD CUP 99 Data Set. These assault classes are as follows in the Table 1:

TABLE I. ATTACK TYPES SORTED BY PROTOCOL

Protocol Type	Attack Name
ICMP	normal, **Smurf**, satan, pod, ipsweep, nmap, and portsweep in addition to portsweep.
UDP	The mentioned terms **include** Teardrop, rootkit normal, as well as nmap and Satan.
TCP	The mentioned terms incluce normal, the Neptune, the guess passwd, the Perl, the land, the ipsweep, the buffer overflow, the ftp writes, and the **loadmodule. satan**, spy, imap, rootkit, warezclient, multihop, warezmaster, phf, and so on.

TABLE II. FUNDAMENTAL PROPERTIES OF EVERY VECTOR OF THE NETWORK CONNECTION

Feature No.	*Name of the*	*Description0*	*Sample of the Data*
1	Connection of Duration	The kind of protocol that is used	0
2	Protocol of type	Type of protocol used	]]UDP
3	destination utilized Service	Utilized in the destination network service	data-UDP
4	Flag.,	A flag signifying the connection to the system	SF

TABLE III. FUNDAMENTAL PROPERTIES OF EVERY NETWORK CONNECTION VECTOR

Feature No.	Feature Name	Description	Sample Data
5	bytes of Source	Transfer of bytes from the source to the destination	700
6	Bytes of Destination	Bytes that were moved from the source to the destination	30
7	Attack of Land	The land assault is described as follows: 1 for true, 0 for false	1
8	wrong of Fragments	Bits in the connection that are either irrelevant or incorrect	0
9	Urgent packets	Urgent bit	0

TABLE IV. CHARACTERISTICS OF EVERY CONTENT-RELATED NETWORK CONNECTION VECTOR

Feature No.	Feature Name	Description	Sample Data
10	Indicators Hot	Signals that are hot	0
11	failed logins Number	Erroneous attempts to log in	1
12	Log status	0: if you have not successfully logged in 1: any other	0

TABLE V. FEATURES OF EACH NETWORK CONNECTION VECTOR THAT ARE RELATED TO CONTENT.

Feature No.	Feature Name	Description	Sample\sData
13	Compromised conditions	Compromised conditions	0
14	Shell root	Root shell 1: otherwise, 0: else	1
15	Attempted SU	As a result, the root command was rendered unsuccessful. If not, 0:	0
16	Root access num	Operations carried out in the root user	5
17	File creations num	Examples of new file creations that have occurred	0
18	Shell prompts num	Quantity of prompts for the shell	1

19	Files access num	The procedures for gaining access to files	0
20	Files access num\southbound cmds num	Commands that are sent out after a session	0
21	Hot login	1: if belongs to hot list0: otherwise	1
22	Guest login	1. Login as a visitor Any other case: 0	0
23	Count0	The number of connections that are made to the same host destination	3
24	Count service	The connections in the service port are the same as the current connection.	4
25	Rate serror	Determine the total number of connections that have activated all four flags (23)	1

TABLE VI. FEATURES OF EACH NETWORK CONNECTION VECTOR RELATED TO CONTENT.

Feature No.	Feature Name	Description	Sample Data
26	Serror_server_rate	The number of connections that have activated all four flags of the system (24)	1
27	Rate _rerror	There are connections that are triggering the REJ (4) flag among the connections (23)	0
28	Rerror_server_rate	There are connections that are triggering the REJ (4) flag among the connections (24)	0
29	Same rate Service	Relationships with the same customer service	1
30	Service rate different	Connections with a variety of service providers	0
31	Service rate different host	Connections with a variety of service providers (24)	0

TABLE VII. TIME-RELATED TRAFFIC CHARACTERISTICS OF EVERY NETWORK LINK VECTOR

Feature No.	Feature Name	Description	Sample Data
32	destination of host count	Connections that have the same Internet Protocol addresses	400
33	host count destination port count for the destination of the host	Those connections that have the same port number	64
34	The destination with the same host service	Connection that is associated with the same service in the host count destination (32)	0.28

TABLE VIII. EACH NETWORK CONNECTION VECTOR HAS TIME-RELATED TRAFFIC CHARACTERISTICS THAT ARE UNIQUE TO IT.

Feature No.	Feature Name	Description	Sample\sData
35	The destination with the same host service	The connections that are related with the various services. Connections constitute the dst host count aggregated total (32)	0.01
36	Destination hosts same\s service port	Connections are grouped together in the Dst host port count variable, and they are all connected with the same source port (33)	0.36
37	diff host differs from destination host srv_	Connection was delivered to a variety of computers belonging to the Dst host port count (33)	1
38	Destination host service s	the links that have been	0
	error	activated the four flags (32)	
39	Destination host service s er	those connections that have the four flags enabled inside them (33)	5
40	The destination host that is rerro	connections with the host count destination flag triggered by the REJ flag (32)	1.07
41	Destination host service r\s error	Those connections that have the REJ bit set on the flag (4). (33)	1

5.2 Metrics for Performance in Intrusion Detection Systems

1.Cost

Cost is a critical component in determining the feasibility of the IDS. IDS expenses may be further categorized using the following three criteria as shown in Table 9.

TABLE IX. CATEGORIZED OF COST

Metric	Description
Damage Cost	In the case that intrusion detection is either difficult to get or insufficient, the cost of damage to the resources that are the focus of the investigation is incurred.
Response Cost	Cost associated with alarming the system or logging intrusions, including system reboot, additional packet filters, recovery, and human resources.
Operational Cost	The resources required for monitoring events, analyzing activities, computing time, and real-time functionality of the IDS.

2. Detection Rate

The efficacy of IDS is gauged by the detection rate. It is possible to assess detection rate in terms of false positives and false negatives display in Table 10.

TABLE X. DETECTION RATE

Metric	Description
False Positive	An alert for an intrusion is triggered even when there is no actual violation of security.
False Negative	Not raising the alert when there has been a real breach.

3. Scalable and Resilient to Attacks

It's possible that the organization utilizing the IDS is always expanding. So, in order to handle this growing traffic, the IDS has to be flexible as shown in Table 11.

TABLE XI. IDS HAS TO BE FLEXIBLE

Scalability	The IDS should be able to handle increasing traffic and adapt to the growth of the institution using it.
Resilience	A dynamic database that can integrate growing signatures and recognize new intrusion patterns should be included in the intrusion detection system (IDS) in order to make it resistant to new forms of assaults.

However, the first instance's findings from the recommended technique showed a higher degree of excellence when compared to the results discovered in the other academic publications examined in Table 12 below.

TABLE XII. COMPARISON BETWEEN PROPOSED METHODS AND PREVIOUS RESEARCHES.

Author	Method	Dataset	Accuracy
Sharma, B et al. [23]	Deep learning-based method	Internet of Things networks	93.05%
Sun, Z et al. [24]	Machine learning	Internet of Medical Things	90.76%
Alwahedi, F.et al. [27]	Machine learning techniques	IoT	92.79%
Liao, H et al. [28]	Deep learning technologies	IoT	94.89%
In our method*	Hybrid Machine learning	IoT	96.36%

6. CONCLUSION

Because the fields of computer science and information technology are expanding at a rapid rate, it is becoming increasingly important to improve scientific and technical activities that are related to computers and the internet. This is because of the fact that these fields are expanding simultaneously. In today's enterprises, society, and personal lives, there is a rising realization that computer networks are becoming more important. This recognition is developing among individuals. This article addresses the question of whether or not tracing and machine learning techniques may be used in the construction of an efficient intrusion detection system. An investigation into the possibility of building an intrusion detector that is based on the activities of connected things is explicitly carried out. The research investigates the intricate relationship that exists between huge flows and the length of time they last in order to determine whether a flow will be short or lengthy. An incremental support vector machine (SVM) approach is used in order to solve issues pertaining to network intrusion detection. The problem of SVM classification might be resolved by using a decision function that takes a quadratic approach to solving the problem. By doing so, it is possible to find a solution to the issue. For the purpose of training an SVM classifier, an incremental support vector machine (SVM) is used in combination with a subset of the dataset that is given. During each step, a support vector is saved, and before to the subsequent iteration, a new training set is constructed. It is essential that the KKD criteria be adhered to in a meticulous manner whenever a new vector is introduced to the training dataset. Furthermore, with the help of particle swarm optimization, SVM parameter refinement may be accomplished in a step-by-step manner. In the event that any of the samples in the training dataset do not fulfill the KKT criteria, it is possible

to construct SVM classifiers for the purpose of network intrusion detection. It is possible, on the other hand, that the SVM classifiers will not be beneficial if the KKT conditions are met.

Conflicts Of Interest

The author's disclosure statement confirms the absence of any conflicts of interest.

Funding

The author's paper explicitly states that the research was self-funded and no support was received from any institution or sponsor.

Acknowledgment

The author would like to thank the institution for creating an enabling environment that fostered the development of this research.

References

[1] A. Megantara and T. Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," Journal of Big Data, vol. 8, no. 1, pp. 1–19, 2023.

[2] G. Perumal, G. Subburayalu, Q. Abbas, S. M. Naqi, and I. Qureshi, "VBQ-Net: A Novel Vectorization-Based Boost Quantized Network Model for Maximizing the Security Level of IoT System to Prevent Intrusions," Systems, vol. 11, no. 8, p. 436, 2023.

[3] A. Mahalingam, G. Perumal, G. Subburayalu, M. Albathan, A. Altameem, R. S. Almakki, and Q. Abbas, "ROAST-IoT: a novel range-optimized attention convolutional scattered technique for intrusion detection in IoT networks," Sensors, vol. 23, no. 19, p. 8044, 2023.

[4] H. Liao, M. Z. Murah, M. K. Hasan, A. H. M. Aman, J. Fang, X. Hu, and A. U. R. Khan, "A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things," IEEE Access, 2024.

[5] K. DeMedeiros, A. Hendawi, and M. Alvarez, "A survey of AI-based anomaly detection in IoT and sensor networks," Sensors, vol. 23, no. 3, p. 1352, 2023.

[6] M. Markevych and M. Dawson, "A review of enhancing intrusion detection systems for cybersecurity using artificial intelligence (AI)," in International Conference KNOWLEDGE-BASED ORGANIZATION, vol. 29, no. 3, pp. 30–37, 2023.

[7] M. Amin, F. Al-Obeidat, A. Tubaishat, B. Shah, S. Anwar, and T. A. Tanveer, "Cyber security and beyond: Detecting malware and concept drift in AI-based sensor data streams using statistical techniques," Computers and Electrical Engineering, vol. 108, p. 108702, 2023.

[8] A. Kathirvel and C. P. Maheswaran, "Enhanced AI-Based Intrusion Detection and Response System for WSN," in Artificial Intelligence for Intrusion Detection Systems, pp. 155–177, Chapman and Hall/CRC, 2023.

[9] R. L. D. Moura, V. N. Franqueira, and G. Pessin, "Cybersecurity in Industrial Networks: Artificial Intelligence Techniques Applied to Intrusion Detection Systems," 2023.

[10] M. Tubishat, F. Al-Obeidat, A. S. Sadiq, and S. Mirjalili, "An Improved Dandelion Optimizer Algorithm for Spam Detection: Next-Generation Email Filtering System," Computers, vol. 12, no. 10, p. 196, 2023.

[11] H. J. Shiu, C. T. Yang, Y. R. Tsai, W. C. Lin, and C. M. Lai, "Maintaining Secure Level on Symmetric Encryption under Quantum Attack," Applied Sciences, vol. 13, no. 11, p. 6734, 2023.

[12] F. Gatica-Neira, P. Galdames-Sepulveda, and M. Ramos-Maldonado, "Adoption of Cybersecurity in the Chilean Manufacturing Sector: A First Analytical Proposal," IEEE Access, vol. 11, pp. 133475–133489, 2023.

[13] E. Debas, N. Alhumam, and K. Riad, "Unveiling the Dynamic Landscape of Malware Sandboxing: A Comprehensive Review," 2023.

[14] K. N. H. De Silva, M. A. S. B. Manchanayaka, D. L. S. I. Punyasiri, H. A. D. N. Perera, A. Gamage, and N. Gamage, "Realtime Network Based Anomaly Detection and Malware Analysis for SMEs and Smart Homes," International Research Journal of Innovations in Engineering and Technology, vol. 7, no. 10, p. 249, 2023.

[15] M. M. Saeed, H. N. R. Mohammed, O. A. H. Gazem, R. A. Saeed, H. M. A. Morei, A. E. T. Eidah, et al., "Machine Learning Techniques for Detecting DDOS Attacks," in 2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA), pp. 1–6, IEEE, Oct. 2023.

[16] S. Sheeja and J. Joseph, "A Three-layer Convolution Neural Network Approach for Intrusion Detection in IoT," in 2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 261–268, IEEE, Nov. 2023.
[17] S. A. M. Al-Rubaye, "Intrusion detection system in IoT networks using SVM-PSO classification," Master's paper, Altınbaş Üniversitesi/Lisansüstü Eğitim Enstitüsü, 2022.
[18] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," Computers and Electrical Engineering, vol. 107, p. 108626, 2023.
[19] A. Verma and V. Ranga, "On evaluation of network intrusion detection systems: Statistical analysis of KDDS-001 dataset using machine learning techniques," Authorea Preprints, 2023.
[20] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," Expert Systems with Applications, vol. 238, p. 121751, 2024.
[21] Z. Sun, G. An, Y. Yang, and Y. Liu, "Optimized machine learning enabled intrusion detection system for internet of medical things," Franklin Open, vol. 6, p. 100056, 2024.
[22] S. Ali, Q. Li, and A. Yousafzai, "Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey," Ad Hoc Networks, vol. 152, p. 103320, 2024.
[23] G. S. R. Emil Selvan, T. Daniya, J. P. Ananth, and K. Suresh Kumar, "Network intrusion detection and mitigation using hybrid optimization integrated deep Q network," Cybernetics and Systems, vol. 55, no. 1, pp. 107–123, 2024.
[24] F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," Internet of Things and Cyber-Physical Systems, 2024.
[25] H. Liao, M. Z. Murah, M. K. Hasan, A. H. M. Aman, J. Fang, X. Hu, and A. U. R. Khan, "A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things," IEEE Access, 2024.
[26] M. Nanjappan, K. Pradeep, G. Natesan, A. Samydurai, and G. Premalatha, "DeepLG SecNet: utilizing deep LSTM and GRU with secure network for enhanced intrusion detection in IoT environments," Cluster Computing, pp. 1–13, 2024.
[27] G. S. C. Kumar, R. K. Kumar, K. P. V. Kumar, N. R. Sai, and M. Brahmaiah, "Deep residual convolutional neural Network: An efficient technique for intrusion detection system," Expert Systems with Applications, vol. 238, p. 121912, 2024.
[28] A. Binbusayyis, "Hybrid VGG19 and 2D-CNN for intrusion detection in the FOG-cloud environment," Expert Systems with Applications, vol. 238, p. 121758, 2024.
[29] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, "Flowtransformer: A transformer framework for flow-based network intrusion detection systems," Expert Systems with Applications, vol. 241, p. 122564, 2024.
[30] M. Saied, S. Guirguis, and M. Madbouly, "Review of artificial intelligence for enhancing intrusion detection in the internet of things," Engineering Applications of Artificial Intelligence, vol. 127, p. 107231, 2024.
[31] U. K. Lilhore, S. Dalal, and S. Simaiya, "A cognitive security framework for detecting intrusions in IoT and 5G utilizing deep learning," Computers & Security, vol. 136, p. 103560, 2024.
[32] J. M. Kizza, "System intrusion detection and prevention," in Guide to Computer Network Security, pp. 295–323, Cham: Springer International Publishing, 2024.
[33] S. Rajasoundaran, S. S. Kumar, M. Selvi, K. Thangaramya, and K. Arputharaj, "Secure and optimized intrusion detection scheme using LSTM-MAC principles for underwater wireless sensor networks," Wireless Networks, vol. 30, no. 1, pp. 209–231, 2024.