SN Computer Science (2022) 3:168
https://doi.org/10.1007/s42979-022-01053-9

ORIGINAL RESEARCH

An Anomaly-Based Intrusion Detection System for IoT Networks Using Trust Factor

Krishan Pal Singh[1] · Nishtha Kesswani[1]

Received: 19 August 2021 / Accepted: 5 February 2022 / Published online: 21 February 2022

Abstract
In recent years, the Internet of Things has grown visibly and soon will be an essential part of our daily lives. This increases the number of transactions in a network and with that risks to these sensitive data increase too; thus, we require a smart system to detect any unauthorized advances to an IoT network and prevent those risks. This system predicts and delivers possibilities of the intrusion based on a few attributes identified using feature engineering. An Intrusion Detection System is tested on its ability to detect malicious activities within IoT networks. Here, we propose an Anomaly-based Intrusion Detection System that detects and prevents attacks on the IoT environment. This approach has two primary objectives to address. First, the data require filtration using the correlation coefficient to combine the probability of distribution to identify features that have a positive impact on the accuracy. Second, the classifier algorithm identifies the behavior using the trust factor based on the selected features. In this step, we analyze the precision, recall, and f1-score of the model on a pre-existing NSL-KDD dataset where the proposed model obtained 98.4% accuracy along with high TPR (True-Positive Rate) and low FPR (False-Positive Rate).

Keywords Internet of Things · Intrusion Detection System · Feature Engineering · Correlation coefficient · Trust Factor

Introduction

IoT allows Internet access to various devices such as smart wearable, smart homes, biosensors, etc., and such rapid growth will make IoT essential to our lives. This rise of IoT in computer networks will bring forth new threats every day; therefore, information security has become the primary concern of the IoT paradigm. An intruder is an entity that causes the above-mentioned concerns, looking to breach a network and exploit it to gain unauthorized access with criminal intentions [1, 2].

We classify intruders into two categories—Masquerader aka outsider, i.e., masking as somebody else by spoofing or stealing credentials and Misfeasor aka insider is someone with limited access to the network, but violates its permissions to misuses the access. Malicious activities in a network are classified into four types as per the KDD dataset:

1. Probe: The attacker scans the network to gain information such as which port is sending packets and which is receiving packets.
2. DoS (denial of service): The attacker sends too many requests to handle for a host machine to interrupt the access of a valid user.
3. U2R (User to Root): The attacker tries to violate the access given as a user (limited user) to gain the root access; this can be done by stealing credentials.
4. R2L (Remote to Local): The attacker gets remote advances to a local user machine. Both R2L and U2R show normal user behavior, and thus are difficult to detect for detection systems.

An intrusion can cause Malware Inclusion, Data Extraction, Network Crash, or Unauthorized access to connected

This article is part of the topical collection "Enabling Innovative Computational Intelligence Technologies for IOT" guest edited by Omer Rana, Rajiv Misra, Alexander Pfeiffer, Luigi Troiano, and Nishtha Kesswani.

✉ Krishan Pal Singh
krishanpal.singh@jnujaipur.ac.in

Nishtha Kesswani
nishtha@curaj.ac.in

[1] Central University of Rajasthan, Ajmer, India

devices in the network thus requiring smart systems to constantly monitor the network traffic and alert the system on the discovery of malicious activities (Table 1). Intrusion Detection Systems are of two types based on their detection mechanisms:

1. Traditional systems are Rule-based and monitor the network traffic with pre-determined rule-sets (Signatures) to dissect benign and malicious traffic. Traditional mechanisms only detect known threats as it consists signature of that attack thus labeled as a signature-based Intrusion Detection System. These mechanisms do not detect 0-day vulnerabilities due to the unavailability of its signature. Traditional methods consist of few limitations as it requires a constant update to rules, but the exponential increase in threat count makes constant updates impractical. Also, IP and source port spoofing diverse the threat as an intruder can perform packet fragmentation to beat the signature-based mechanisms.
2. Next-Gen or Anomaly-based Intrusion Detection System copes with new emerging threats and detect 0-day threats. This variant uses Artificial Intelligence and especially Machine learning to revolutionize threat detection and prevention. Anomaly-based methods are scalable and their debugging is a sophisticated affair due to features such as real-time detection accuracy and target-oriented tunability. Anomaly-based mechanisms have three phases. First, Training of Data allows to write and modify the rules according to the data. Second, Machine-learning Model uses the trained data into a classifier, i.e., Decision tree, Artificial neural network, etc. to predict the behavior. Third, objectives of the model, i.e., Accuracy, Recall value, True and False Positive or negative rate, f1-score and Precision, etc.

Network loopholes or breaches are increasing in number and size exponentially, and with it, the cost of damage is increasing too. Size of Intrusion-a renowned finance firm in USA had been breached in 2019 with around 800 million records of the firm were leaked online that included financial transactions, SSN (Social Security Numbers) of its customers, and a lot more. Growth in number of intrusions in between 2017 and 2019, there has been an 80% rise in the number of patients affected by these breaches in health industries. 4 million USD is the global estimated average cost of intrusion damage. Various categories of detection techniques are as follows [3]:

Statistical analysis: This method compares the current set of data with the normal behavior of the network. This is generally used in anomaly-based detection mechanisms.

Evolutionary algorithm: A similar to statistical analysis model with the only deviation lies with their ability to differentiate attempted intrusions and normal behaviors using model dissection based on various conditions.

Protocol verification: It establishes standards of normal behaviors and compares them to the current flow. Data that violate the pre-determined standards are labeled malicious. This mechanism is successful in commercial environments, but it has the limitation of incorrectly judging false positives for unspecified protocols.

Rule-Based: It compares each flow with its signatures, and if found different, then it will label the flow as suspicious.

Artificial Neural Network: In this mechanism, complex hypotheses are formed using neurons with complexity directly proportional to the number of neurons. The hypotheses are evaluated by arranging the nodes as input for feedback.

Classifications of intrusion detection systems [4]:

i. Network: Network IDS are planted at a specific point in the network itself to monitor traffic. It examines the traffic on all of the subnets and compares it to all the known threats. If a threat is found or the traffic of the subnet is somehow abnormal, an alert is sent to the admin.
ii. Host: In Host IDS, an independent node(Host) is identified and selected to run HIDS on the network. It observes the packets arriving on and departing from a node and an alert will be sent to the admin if any abnormal behavior is detected. A snapshot of the current status of the node is captured and compared to the previous snapshot. Alteration or deletion of crucial system files will trigger an alert to the admin to investigate.
iii. Protocol-based: Here, the system is supervised by an agent at the front-end of a server, managing, modifying, and translating the protocol between a node and the server to secure the server by constantly observing the flow of the HTTP.
iv. Application Protocol-based: In this IDS, an agent is invoked among a group of servers that identifies the

Table 1 Labels in NSL-KDD datasets

Type of label	Number of occurrences
Probe	45268
DoS	4114855
U2R	297
R2L	15676
Normal	978834
Total	5209460

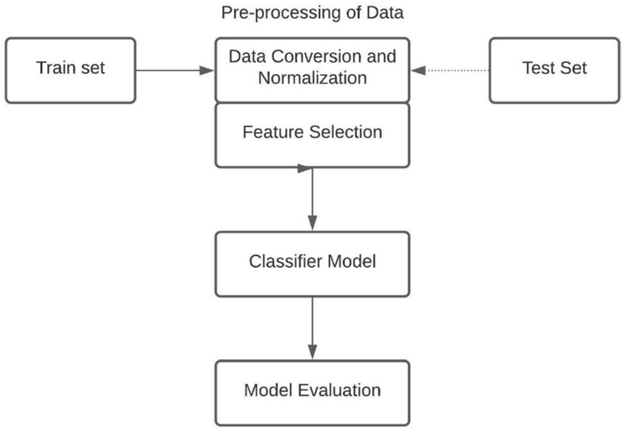

Fig. 1 Generic Flow of Pre-processing of Data

malicious activities by observing and translating the stream on app-specified protocols.

v. Hybrid: Hybrid is a combination of multiple IDS mechanisms to develop an entire image of the network to observe and monitor the protocol streams as well as the traffic flow among the nodes.

Challenges of developing an anomaly detection mechanism (Fig. 1) [3]:

Heterogeneity: Heterogeneity is a huge security roadblock for IoT devices, because different benign data patterns to each device with dissimilar surveillance objectives make it difficult to train a single accurate anomaly Detection model for all of the networks.

Detection Accuracy: Anomaly detection mechanisms are classified into non-parametric and parametric mechanisms. The parametric mechanism prioritizes the data distributions, whereas non-parametric methods allow vague data patterns to locate the areas with minimal data density; hence, the detection accuracy shifts to be a secondary objective.

High Computational Complexity: The complexity increases due to high data features, referred to as the curse of dimensionality.

Untrustworthy networks and devices are inevitable; hence, the search for malicious adversaries within a network is a fundamental requirement. Since the IoT environments function with storage and power constraints, these detection for adversaries need to be optimal to resource requirements and computability. Furthermore, Intrusion detection in IoT networks offers great prospects for some much-needed research that results in various mechanisms that detect potential vulnerabilities and attacks such as Probe, U2R, R2L, MITM, and DOS. IDS has to be reliable to protect an IoT network from such threats and alerts if such an incident occurs.

In this paper, an Anomaly-based intrusion detection mechanism is proposed based on the Decision tree mechanism with a mathematical approach to detect anomalies using the trust factor.

Major contributions of the proposed mechanism are as follows:

1. Initial dimension reduction is performed using correlation.
2. Furthermore, we introduced an add-on feature of information origin.
3. We propose a novel idea to evaluate Trust using a combination of both the selected features and the add-on feature for each packet (Fig. 2).

This trust factor will determine whether the packet is benign or malicious. This new model aims to achieve maximum possible accuracy ailing minimum possible resources for its fluency in low power networks such as IoT networks.

In this section, the definition and the outright necessities of the Intrusion Detection System are stated. The section “Related Work” involves the related work that consists of the information gathered via the study of various Research Papers, E-books, and Articles. The section “Classification Matrix” explains the Classification Confusion matrix. The section “Overview of Proposed Model” introduces the proposed idea of the mechanism in detail and the section “Experimental Setup and Result Analysis” consists of the experiments performed to achieve the objectives of the proposed mechanism. The section “Conclusion and Future Work” represents the conclusion of the work and the future scope.

Related Work

Dorothy E. Denning introduced the first-ever intrusion detection system to detect pre-determined intrusions based on a dual approach of rule-based and statistical anomaly detection. This solution model known as the intrusion detection expert system [5] was released by SRI international to detect intrusion based on user profiles, host, and target systems. Lincoln Laboratory at MIT presented the idea to detect anomalies with DARPA Evaluation (1998–99) [6], but J. Mchugh showed DARPA’s lack of performance, while actual network simulation in [7] thus a need for new datasets emerged to develop IDS.

Eduardo et al. [8] combined statistical methods and self-organizing maps to detect anomalies in a network for classification of benign and malicious behaviors, using Fisher’s ratio of discrimination and PCA for Feature selection. Ujwala et al. [9] presented a hybrid mechanism combining various data mining methods. The attributes at each data point are filtered using the K-means clustering algorithm.

Haddadpajouh et al. [10] introduced a two-layer detection mechanism to address the high dimensionality issue involving both supervised and unsupervised Dimension Reduction methods (Fig. 2). Principal Component Analysis performs feature engineering, while Linear Discriminant Analysis speeds up the detection process. Features in the PCA-build dataset cannot be categorized, and because of that, LDA reduces the features to transform the data into new dimensions. This model gains high performance with low resources, and thus is deployed to detect intrusion attempts and residual attack patterns in IoT networks for digital forensics. This mechanism is also efficient to detect and handle U2R and R2L attacks due to the low frequency and low profile of these attacks.

Murali et al. [11] proposed a lightweight algorithm to prevent Sybil attacks that required minimal computation with high accuracy using an Artificial Bee Colony (ABC) model for Mobile RPL in the IoT Networks. This algorithm is inspired by Honey bees who have a limited flight range, and within that, they build nectar and forage(Pollen Collection process) to achieve maximum honey production and breeding, bees locate the best hive establishment point. Their Forage depends on four key factors nectar, Collector bees, Observer bees, and Locator bees. ABC Algorithm is a population-based algorithm for optimal simulation of forage of honey bees. In terms of the algorithm, collector bees are compromised Sybil identities, nectar is a collection of compromised identities, locator bees are the main attacker, and observer bees are the compromised nodes that look to compromise the nearby neighboring nodes.

Algarny et al. [12] proposed a suggestion to use a Bayesian network to improve the detection rate of R2L attacks by conducting experiments with various attributes of the KDD99 dataset and achieved an accuracy of 86%. Dewangan et al. [13] suggested assembling a Bayes net and ANN to separate benign and attack classes in NSL-KDD and achieved an accuracy of 98% using 35 attributes. Waghmare et al. [14] and Dhanabal et al. [15] used support vector machine (SVM) to introduce a method to dissect various attacks with different RBF kernel functions to gain high classification accuracy of 98% using k-fold cross-validation.

Classification techniques predict by analyzing the available data [15]. A few of these techniques involves Decision tree, Random Forest, SVM, Neural Networks, and KNN. Decision Tree divides the problem into multiple sub-problems creating a decision tree to come up with a model for classification, Neural networks refers to a neural driven set of stats learning models to approximate functions that depend on a usually large dataset for the training of the model and Nearest neighbor classifies new subsets based on similarity or correlation measure using previous subsets. All the aforementioned models are represented inherent limitations and unique features. A decision tree requires more time to build, whereas if the size dataset increases, the KNN method becomes gradually time-consuming and ANN performs well if data are only numerical, thus requiring the encoding of the textual attributes into numerical data.

Wagh et al. [16] and Qiu et al. [17] discussed various Machine Learning-based detection mechanisms in their survey and stated the pros and cons of each model, respectively. Vokorokos et al. [18] developed an IDS based on multi-layer perceptron for SLPs and Planquart [19] developed a similar model based on self-organizing maps for unsupervised techniques. Sahu et al. [20] used neural networks to improve the accuracy of detection efficiently for anomaly detection and misuse detection, and [21] assessed the performance of their IDS using various existing datasets. Surveys in [22] displayed the difficulties modern IDS discover while handling high-speed traffic in the network. Researchers prompted the advantage to attackers while exploiting these weaknesses in high-speed traffic by overloading an IDS. Kumar et al. [2] proposed a mechanism to detect the network activities of malware in enterprise-level and Internet Service Provider networks with a majority of current IoT malware are sectioned into limited categories to identify similar malware and to ease the task to identify detection methods for them. The traffic patterns of malicious behavior from each section are monitored through packet capturing and testbed experiments and a solution EDIMA (Early Detection of IoT Malware Network Activity) to detect the malware activity using Machine Learning techniques is presented with the analyzed traffic patterns.

Gajewski et al. [23] proposed an approach to use traffic data from Home Gateways and compare the results achieved from each Home Gateway to detect global anomalies and ensure improved security by observing the communication patterns between terminal nodes and the Home Gateways and partitioning the process of anomaly detection among two objects, the access devices and the data center. This detection and interpretation process is dissected between the network operator and the end-users where the operator might utilize all of its resources to store, analyze, and decide whether suspected actions and addresses are a source of an attack to provide additional security features to protect the network against cyber threats. This method does not provide

Table 2 Classification matrix

Actual	Prediction	
P	P	True Positive
N	P	False Positive
N	N	True Negative
P	N	False Negative

security against intrusion detection done using other methods, i.e., Deep Packet Inspection for matching patterns based on protocol type, Internet addresses, ports, etc.

Classification Matrix

In Table 2, True Positive (TP) represents classification occurrence of benign data as benign data and False Positive (FP) represents the occurrence of classification of benign data as abnormal data, whereas True Negative (TN) shows the classification of abnormal data as abnormal and False Negative (FN) represents the classification of abnormal data as benign data.

Detection accuracy is a ratio of the total number of benign predictions made

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}. \tag{1}$$

True-Positive Rate (TPR) is a function of the total number of correct predictions of benign data, also known as recall, whereas False-Positive Rate (FPR) shows an incorrect diagnosis of malicious cases as benign and True-Negative Rate (TNR) represents abnormal cases correctly classified as abnormal, while False-Negative Rate (FNR) is a ratio of incorrect diagnosis of benign data as malicious data

$$\text{Recall, or, TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{2}$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}; \text{ FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}}. \tag{3}$$

Precision is represented as positive prediction ratio

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{4}$$

f1-score seeks balance between precision and recall

$$f1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

Overview of Proposed Model

An effective intrusion detection mechanism has high precision and performance and achieves these objectives we involve the following steps.

Pre-processing of Data

1. Identify a proper dataset with good quality of data, i.e., NSL-KDD.
2. Split the dataset into training and testing data in about 4:1 ratio.
3. Training data will be dissected into two further subgroups, Training set and Validation set.
4. The pre-processing phase enables the filtration of the data to eliminate insignificant information. The proposed model identifies an established root for initial discrimination of data using a correlation coefficient to partition the features into groups. The second level of reduction is done using a low-variance filter mechanism to extract the features that rationalize the output values to make new sets of features, and then, an add-on feature of information origin is involved to calculate a trust factor using that add-on and remaining attributes.
5. The proposed model will be trained on the training dataset, and then, it will be evaluated on the validation set with major parameters being the training accuracy of the model.
6. After Validation, the model will be tested on the test dataset and judged based on testing time and accuracy of the model. The performance of the model on the test data will provide the predicted accuracy and efficiency of the model in a real-time scenario.

Feature Engineering

Feature Engineering is an essential phase of modeling that showcases the selection process of acute inputs for an ML model. Two unique sets of features selected independently would provide different outcomes of detection; thus, feature identification is influential for the accuracy, efficiency, and range of an IDS. A few objectives of Feature Engineering are as follows.

To Reduce the Dataset: Avoid unnecessary calculations by opting for features that deviate the output significantly.

To Improve the Accuracy: Remove features with no lead or optimal relations for less calculation overhead.

Avoid Excessive Noises: If an ML model states defects or noises instead of the defined relations, it is called overfitting. It occurs due to complex models, having poor predictions that lead to fluctuations in the data. Avoid selecting co-related features to reduce overfitting.

Description of the Dataset

KDD Dataset: A benchmark for research in the domain of Intrusion Detection techniques, KDD still has ample research going on this dataset to enhance the performances of IDSs. These studies use KDD to train and test their respective detection mechanisms. The KDD dataset can be dissected into four groups that are Basic, Traffic, Content, and Host according to their data attributes. KDD features two primary evaluation metrics, False Detection Rate (FDR), and True Detection Rate (TDR). As a result, we understand that to enhance the performance of the dataset to reach maximal TDR with minimal possible FDR [24].

Though the latest updates of this dataset still consist of few traditional issues such as redundancy and do not represent the existing real-world networks due to the absence of the public datasets for NIDSs, yet KDD is still vastly applied as a benchmark by researchers to compare various mechanisms because of the large set of records, KDD contain to train and test new mechanisms and it is affordable to perform tests on the entire dataset without selecting a small section of data randomly leading to outcomes of various works comparable and consistent (Table 3) [25].

Table 3 Distribution of labels in training and test datasets

	Training size (%)	Test size (%)
Probe	1.01	0.4
DoS	79.98	79.57
U2R	0.001	0.002
R2L	0.1	1.2
Normal	18.90	18.82
Total records (in number)	4167568	1041892

Train the Classifier, Predict the Behavior, and Analyze Performance of the Proposed Model

1. Calculate Co-relation Coefficient(r): Pearson Co-relation coefficient is used to calculate the value of the coefficient. Here, a & b are relational scores and N is the number of paired relations

$$r = \frac{N \times \epsilon ab - (\epsilon a) \times (\epsilon b)}{\sqrt{[N \times \epsilon a^2 - (\epsilon a)^2] \times [N \times \epsilon b^2 - (\epsilon b)^2]}} \tag{6}$$

   if $(0 < r < 1) \rightarrow$ Positive relation between a & b that means if a increases, b increases as well. else if $(-1 < r < 0) \rightarrow$ Negative relation between a & b that means if a increases, b will decrease. if $(r = 0) \rightarrow$ No relation between a & b.
2. Sort all co-related features into a group: if any feature appears in multiple groups, then select that feature only once in any of the groups and make it redundant.
3. Select one of the co-related features: if two or more features are related to each other, then select one of the aforementioned features and eliminate the other features, i.e., Filtration with correlation coefficient leads

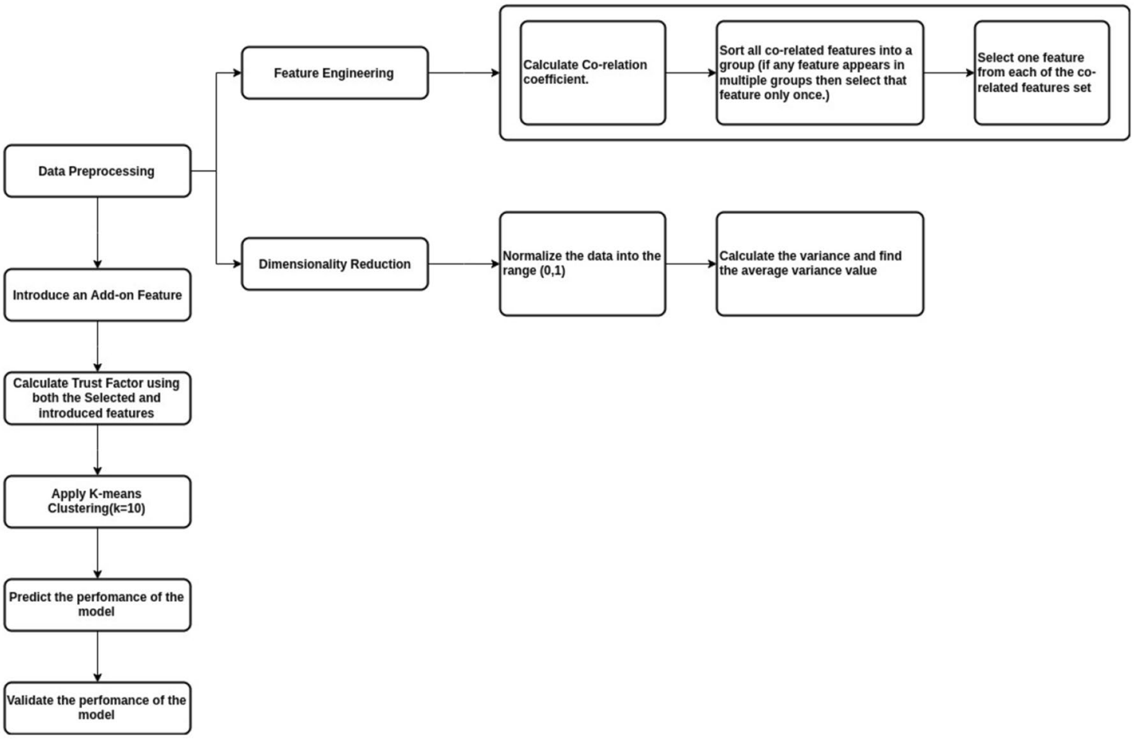
Feature Engineering
Calculate Co-relation coefficient.
Sort all co-related features into a group (if any feature appears in multiple groups then select that feature only once.)
Select one feature from each of the co-related features set
Data Preprocessing
Dimensionality Reduction
Normalize the data into the range (0,1)
Calculate the variance and find the average variance value
Introduce an Add-on Feature
Calculate Trust Factor using both the Selected and introduced features
Apply K-means Clustering(k=10)
Predict the perfomance of the model
Validate the perfomance of the model

Fig. 2 Flowchart of the Proposed Model

Table 4 Selected features and its description

Feature	Feature symbol	Description
4	Service	used service at destination network
5	Flag	connection status (normal or error)
6	src_bytes	Total number of Transferred databytes in a single connection from source to destination
7	dst_ bytes	Total number of Transferred databytes in single connection from destination to source
12	num_ failed_ logins	Count of failed login attempts
27	srv_ serror_rate	Connections that activate the flags among the aggregate connections in srv_count
30	same_ srv_ rate	Ratio of same services connection among the aggregate connections count
31	diff_ srv_ rate	Ratio of different services connection among the aggregate connections count
32	dst_host_srv_count	Ratio of same services connection among the aggregate connections dst_host_count
36	Dst_host_same_src_port_rate	Same source port connections ratio among the aggregate connections in dst_host_srv_count
37	Dst_host_srv_diff_host_rate	Different Destinatiom host system connections ratio among the aggregate connections in dst_host_srv_count
Add-on	info_origin	States the origin of the information

us to 19 subgroups from which we use a feature each from every subgroup.

4. After elimination Normalize the selected feature list and phase into rule-based selection criteria to further remove useless features.
5. Introduce an add-on Feature of information origin and to get the information origin feature of the data, we use various tools, i.e., download managers, package managers, and sandboxing tools such as Any.run by keeping tabs on all the packets entered in the network, i.e., phishing, application exploitation, or download driven and also look to identify penetration attempts by malicious packet before being triggered into the network, because most of the malicious packets upload their abnormal code segments to steal the intellectual property of a network using spoofing or fragmentation thus makes detection difficult.
6. Encode the data from text into numerical and Normalize it in the range [0,1] to calculate the weighted sum of the variance of each information and find average variance value

$$W_i = \frac{\sum[i \times (X-\mu)^2]}{n} \ ; \quad \mathit{for}\ i\epsilon[1,n], \tag{7}$$

where n = number of remaining samples,
μ = Mean of the samples,
X = Random sample.
W = Weighted sum of the variance of the sample.
.

7. Trust is a phenomenon of a belief or approximation to subjectively estimate the future behavior of an entity with other entities. Trust contains a few constant characteristics, i.e., subjectivity, transitivity, Awareness of the context, time decomposition, and measurability. Traditional Methods for trust estimation quantify trust with characteristics that affect trust. These models involve The Bayesian method, weighted average estimation mechanisms, subjective logic, fuzzy logic, and game theory. Since most of these methods are mathematical approaches and hard to implement, we introduce a simple but effective method for trust evaluation. The trust-related feature selection influences the accuracy of the model vastly.
8. Calculate a trust factor for selected information with both dynamic and add-on features using

$$\text{Trust Factor}(T_f) = f[\text{info\_origin} \oplus (W_i,\ n)] \times \frac{\sum W_j}{R}. \tag{8}$$

Here, i = Selected feature sample,
j = selected record sample,
R = Total number of records,
n = total number of remaining features.

9. Set a threshold using the trust factor. A high value of trust factor indicates normalcy of that node, and as the trust factor value decreases, it denotes the possible amount of impurities in that node.
10. Use k-fold cross-validation to predict the performance of the proposed model(for k = 10). {We generated data from multivariate normalization with different variances. The goal is to group data points into distinct non-overlapping subgroups to identify an ample amount of clusters, so that the data can learn itself. Also, it is necessary to identify adequate assumptions behind an algorithm to have a strong idea of the strength of the method (Table 4). Hence, we opt k =10, so that we have ten groups of data where each group is generated from different normalization.}

Table 5 Resource requirement for experimental setup

Resource type	Resource detail
Operating System	Ubuntu-64 bit v20.04
Primary Memory	8GB
Virtualization Environment	Virtual Machine Manager v2.2.1
Programming Language	Python v3.0
ML libraries	Numpy, sklearn, pandas, Matplotlib, & Tenserflow
Datasets	KDD

Table 6 Accuracy in detection

Class Label	Training Accuracy	Test Accuracy
Probe	99.1	98.6
DoS	99.3	98.9
U2R	98.8	97.2
R2L	98.2	97.8
Normal	99.7	99.5

Table 7 Precision, recall, and *f*1-score

	Training	Test
Precision	0.97	0.94
Recall	0.98	0.96
*f*1-score	0.95	0.98
Response time	18.8467	0.372425

11. Monitor the test set performance of the model. Calculate Accuracy, Precision, recall, and False-Positive Rate (FPR).

Experimental Setup and Result Analysis

In this section, the outline of the experimental setup is provided, i.e., used to analyze the performance of the proposed model such as precision, recall, accuracy, FPR, and the response time of both training and test sets to monitor the efficiency of the model. First, We enlist all the required resources for the experimentations (Table 5).

Redundancy Rates and Detection Accuracy in Training and Test datasets

Redundancy stats of Training set: From 3,379,897 original records of attacks, 3,346,774 distinct records of attacks are eliminated with reduction rate up to 99.1%.

Table 8 Detection accuracy and response time of various previous model in comparison to the proposed model

Model name	Training time	Testing time	Training accuracy	Testing accuracy
Gaussian NB	1.6164	0.2473	88.01	87.95
DT	2.7960	0.0264	99.05	98.06
RF	24.3199	0.6469	99.79	99.63
SVM	804.9260	45.1130	99.87	99.83
LR	82.0133	0.2841	99.35	98.80
Proposed Model	18.8467	0.3724	99.10	98.40

Redundancy stats of Test set: From 796,189 original attack records, 783,449 distinct attack records with reduction rate up to 98.4%.

Table 6 exhibits that the proposed mechanism has a high detection rate over all existing class label and Table 7 provides a brief scenario of high true-positive rate of the proposed model.

Conclusion and Future Work

While evaluating the proposed mechanism in the training set, we evaluated that this model has achieved a redundancy rate of 99.1% with a prediction accuracy of 99.02%. Table 7 shows the high performance of the proposed model in training. The detection accuracy in the test set is 98.40%. The Trust factor threshold set in the domain of (0,1) has the optimal value of 0.7 (Table 8). The proposed model achieved high TPR (0.96) and low FPR (0.2), respectively. The behavior analysis of the KDD dataset is performed using the tables to allow the observer to have a clear understanding of the dataset and its features.

Since the age of Artificial intelligence is approaching fast and the security of the information and the networks, to be the forte of a huge number of studies, we would attempt to improve the accuracy and the precision of this mechanism to paves some ground towards confidentiality of the communication and the security of the networks. For further studies, we suggest the probability to use Ton_IoT dataset as a benchmark to further expand this domain and implement the proposed mechanism in a fully functioned distributed Network.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Singh KP, Tomar S, Jain U, Hussain M. Simplified and secure session key sharing for the Internet of Things (IoT) networks. In: International conference on internet of things and connected technologies. Berlin: Springer; 2020. p. 319–32.
2. Kumar A, Lim TJ. EDIMA: early detection of IoT malware network activity using machine learning techniques. IEEE World Forum Internet of Things (WF-IoT). 2019. https://doi.org/10.1109/WF-IoT.2019.8767194.
3. Chang H, Feng J, Chaofan D. HADIoT: a hierarchical anomaly detection framework for IoT. IEEE Access. 2020. https://doi.org/10.1109/ACCESS.2020.3017763.
4. Depren O, Topallar M, Anarim E, Ciliz MK. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst Appl. 2005;29:713–22. https://doi.org/10.1016/j.eswa.2005.05.002.
5. Denning DE. An intrusion-detection model. In: IEEE Symposium on Security and Privacy, 1986, pp. 118–131.
6. Lincoln M. Laboratory, DARPA intrusion detection data sets. 2016. https://www.ll.mit.edu/ideval/data/. Accessed 07 Apr 2016.
7. McHugh J. Testing intrusion detection systems: a [33] critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Trans Inf Syst Secure. 2000;3(4):262–94.
8. De la Hoz E, De La Hoz E, Ortiz A, Ortega J, Prieto B. PCA filtering and probabilistic SOM for network intrusion detection. Adv Comput Intell. 2015;164:71–81.
9. Ravale U, Marathe N, Padiya P. Feature selection based hybrid anomaly intrusion detection system using K-means and RBF kernel function. In: Proceeding of international conference on advanced computing technologies and applications, ICACTA-2015, procedia computer science, vol. 45. Amsterdam: Elsevier; 2015. p. 428–35.
10. Haddadpajouh H, Javidan R, Khayami R, Dehghantanha A, Choo K-KR. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. IEEE Trans Emerg Top Comput. 2016. https://doi.org/10.1109/TETC.2016.2633228.
11. Murali S, Jamalipour A. A lightweight intrusion detection for Sybil attack under mobile RPL in the Internet of Things. IEEE Internet Things J. 2019. https://doi.org/10.1109/jiot.2019.2948149.
12. Altwaijry H, Algarny S. Bayesian-based intrusion detection system. J King Saud Univ Comp Inf Sci. 2012;24:1–6.
13. Shrivas AK, Dewangan AK. An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set. Int J Comput Appl. 2014;99:8–13.
14. Bhavsar YB, Waghmare KC. Intrusion detection system using data mining technique: support vector machine. Int J Emerg Technol Adv Eng. 2013;3:581–6.
15. Dhanabal L, Shantharajah SP. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int J Adv Res Comput Commun Eng. 2015;4:446–52.
16. Wagh SK. Survey on intrusion detection system using machine learning techniques. Int J Comput Appl. 2013;78(16):30–7.
17. Qiu C, Shan J, Polytechnic B, Shandong B. Research on intrusion detection algorithm based on BP neural. Network. 2015;9(4):247–58.
18. Vokorokos L, Balaiz A, Chovanec M. Intrusion detection system using self-organizing map. Informatica. 2006;6(1):1–6.
19. Planquart J-P. "Application of neural networks to intrusion detection". https://rr.sans.org/intrusion/neural.php (2001).
20. Sahu SK, Sarangi S, Jena SK. A detailed analysis on intrusion detection datasets. In: Souvenir 2014 International Advance Computing Conference IACC, 2014, pp. 1348–1353.
21. Allen J, Alan C, William F, John M, Jed P. State of the practice of intrusion detection technologies, Technical Report no. CMU/SEI-99-TR-028, Tech. Rep., 2000, p. 221.
22. Paxson V. Bro: a system for detecting network intruders in real-time. Comput Netw. 1999;31(23):2435–63.
23. Gajewski M, Batalla J, Mastorakis G, Mavromoustakis C. Anomaly traffic detection and correlation in smart home automation IoT systems. Trans Emerg Telecommun Technol. 2020. https://doi.org/10.1002/ett.4053.
24. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the kdd cup 99 data set. In: Proceedings of the Second IEEE international conference on computational intelligence for security and defense apps. Piscataway: IEEE Press; 2009. p. 53–8.
25. Choudhary S, Kesswani N. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. Proced Comput Sci. 2020;167:1561–73. https://doi.org/10.1016/j.procs.2020.03.367.