International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 7 ǁ July 2022 ǁ PP. 96-103

IoT: False Positive Rate Reduction in Intrusion Detection Systems

Bharti Rana[1], Ravleen Singh[2]
Central University of Jammu[1], Madhav University[2]

Abstract: *The Internet of Things (IoT) is a cutting-edge concept that integrates the physical world with the internet to significantly advance computer technology. It will be extremely challenging to develop transparent health policies and implement a specific security strategy due to the low openness and lack of contact with a substantial number of these devices across a vast area of size. Additionally, in a deliberate attempt to destabilize the business, IoT networks become incredibly silent and defenseless. The IoT should ideally have more clear security tools. The Intervention Access Program can help with this (IDS). The topic of IDs has seen a lot of research, but there are still several important problems that need to be solved. IDSs are capable of detecting several types of interference with a lower proportion of false positives and abnormalities. Malicious users from all over the world are also drawn in by this improvement in performance. Therefore, a more secure environment devoid of all kinds of disruptive agents is required for IoT. Researchers once concentrated on identifying human signatures using signatures and confusion. Zero-day attacks, however, have made it impossible to use these methods for detection. As a result, an IDS model based on the Random Forest (RF) section is proposed in our research. Accuracy, detection rate, and false alarm level are considered when evaluating model performance using the NSL-KDD data set. The suggested model has a Positive False Rate of 0.50 percent and an average accuracy of 99.4 percent and 98 percent for binary and multi-stage classification, respectively. The accuracy of the suggested model was compared with other comparable techniques in the literature, demonstrating its efficacy. We have utilised the Google Colab environment for testing purpose.*
Keywords: *Internet of Things, Intrusion Detection Systems, Random Forest, Ensemble Learning*

Date of Submission: 24-06-2022 Date of acceptance: 05-07-2022

I. INTRODUCTION

A hostile operation that exposes sensor nodes to attack is referred to as an intrusion. A network security system that detects unauthorised traffic is known as an intrusion detection system (IDS). As a second line of defence, it can also protect the network from intruders and various types of attacks such as physical layer attacks, network layer attacks, software attacks, and data attacks. Node tampering, injecting malicious nodes, sleep denial, and side-channel attacks are examples of physical layer attacks. Traffic analysis, spoofing, sinkhole attacks, replay attacks, and DoS are examples of network layer attacks[1]. Viruses, worms, spyware, and malware are examples of software attacks. Data breaches, inconsistent data formats, and unprivileged access are all examples of data attacks. An intrusion detection system (IDS) can be either software or hardware.

IDSs are able to keep an eye on user activity and computer activity, as well as identify the signatures of well-known attacks and different types of dangerous network traffic. The purpose of IDS is to inspect businesses and hubs, recognise various forms of intrusions inside the enterprise, and alert customers to intrusions. The IDS serves as a warning system or organisational eyewitness, preventing damage to the structures by sounding an alarm before the attackers begin an attack. It can distinguish between attacks from the inside and outside.

Interior assaults are initiated by nefarious or negotiated centres that are important to the organisation, as opposed to exterior assaults, which are launched by outsiders who are started by an external organisation. IDS recognises the company's packages and evaluates if they are from legitimate customers or intruders. Three components of ID are Monitoring, Analysis, and Identification, in addition to Alarm[2]. The company's transactions, use cases, and assets are examined in the observing portion. According to a predetermined calculation, the Examination and Detection module of IDS locates disruptions. The caution section sounds an alarm if an interruption is found [3].

IDS methods can be divided into four categories: hybrid, anomaly-based, signature-based, and specification-based. Pattern matching techniques are used in Signature-based IDS (SIDS) to find known assaults; this process is also known as Misuse Detection or Knowledge-based Detection. Matching techniques are used to find a prior incursion in SIDS. In other words, when an intrusion signature resembles a previous incursion whose signature is still there in the signature archive, a notice alert is created. SIDS scans the host's

logs for commands or actions that have previously been labelled as malware. SIDS is also referred to as Knowledge-Based Detection or Misuse Detection in the literature. It might be sufficient to obtain signature information from numerous packets given the complexity of contemporary malware[4]. The contents of earlier packets would also need to be carried by IDS. In general, there are several ways to create SIDS signatures, and these signatures are typically created as formal language string patterns for state machines [5].

Because it can get around the restrictions of SIDS, anomaly-based IDS (AIDS) has drawn a sizable number of academics. An anomaly is defined as a significant deviation from the model, which is analogous to interference, in observable behaviour. This strategy makes use of the distinction between malicious behaviour and typical user behaviour. The definition of an intrusion is anomalous user conduct that deviates from typical user behavior[6]. Utilizing AIDS has certain benefits. They might discover internal harmful behaviour as a start. An alarm is triggered when a hacker makes deposits in a compromised account that can be linked to regular user activity. Second, a cybercriminal may find it challenging to discern what constitutes legitimate usage of the system because it is made up of customised accounts. Second, because the system is made up of individual accounts, it can be challenging for hackers to identify user behaviour that won't result in a security alert[1]. The main difference is that SIDS can only identify intrusions that have already been discovered, whereas AIDS can detect zero-day attacks. AIDS may cause high FPR, though, as deviations might just be brand-new regular events as opposed to actual incursions.

IDS with specifications (SIDS) The specification is a set of guidelines and requirements that describe the appropriate behaviour for network elements such as nodes, routing tables, and protocols. When network activity deviates from specification definitions, intrusions are detected via specification-based approaches[7]. As a result, specification-based detection serves the same objective as anomaly-based detection, which is to identify departures from the norm. However, there is a key distinction between the two methods: in specification-based systems, each specification's rules must be manually established by a human expert. Manually set requirements often have lower false-positive rates than anomaly-based detection. Furthermore, since they might begin operating as soon as the standard is issued, specification-based detection systems don't require any training. On the other hand, manually developed standards [8] may not be flexible enough to fit different situations, take time, and are prone to error [5]. The Hybrid method, as its name implies, combines several different techniques to include a framework for identifying high impacts. Hybrid systems are designed to maximise the positive aspects of existing techniques while minimizing their negative aspects [1].

1.1 Contribution

- The detailed working of the Random Forest classifier is thoroughly discussed.
- An IDS is proposed by employing a Random Forest classifier to reduce the false-positive rate.
- Finally, the proposed model is assessed with the existing models based on key performance parameters.

II. RELATED WORK

A two-tier categorization model and network anomaly-based IDS were proposed by Pajouh et al. [9] and are being used in IoT networks. The authors focused on two different attack types, U2R and R2L, and showed an outstanding true positive rate with low FPR owing to the addition of a refinement function and decreased computational cost. The authors used both unsupervised and supervised reduction methods, including Principal Component Analysis (PCA), which was used for feature extraction and selection, and Linear Discriminant Analysis (LDA), which was used for IDS.

Initially, Meidan et al.'s[3] deep learning autoencoder-based anomaly detection engine was evaluated against Mirai and BASHLITE botnets, two of the biggest botnets. For IDS placement, the authors employ a hybrid strategy in which a central unit communicates with system-level encoders (each encoder is responsible for profiling individual IoT devices). Similar to this, Kasinathan et al. [10] proposed a centralized system with the primary objective of identifying Assaults in 6LoWPAN-based networks. Suricata, a well-known signature-based algorithm, was modified by the authors for use in 6LoWPAN networks. Based on the study a DoS defense controller does after receiving an alert from an IDS, an attack is confirmed. To reduce the number of false positives, a general analysis is employed.

In [11], a deep learning model combined with intrusion detection methods was introduced. The modeling framework for the migration learning and feature extraction process is described in this research. Ten percent of the data from the KDD CUP 99 data set was used as training data. Experimental results show that the suggested approach has a faster detection time and better detection performance. Contrarily, Pacheco et al. [1] proposed a method for creating an intrusion detection system in the event of an od sensor breach. With a high detection rate and few false alarms, the method can accurately authenticate sensors based on their activity and identify both known and unidentified sensor attacks. To precisely categorize normal, an s-DNA profile was developed. As part of the process for analyzing anomalous activity, an s-DNA profile was developed to

precisely categorize typical sensor actions. A methodology for categorizing assaults was also developed, with an accuracy rate of 97.4 percent for unknown attacks and 98.8 percent for known attacks.

To investigate any network penetrations, Tabash et al. [12] offer a hybrid intrusion detection model using Naive Bayes and deep learning techniques. This idea is divided into two parts. The first step comprises extracting, discretizing, and reducing dimensionality using the Genetic Algorithm (GA). After merging the Nave Bayes classifier (NB) and the Decision Table after the first stage (DT), the second stage is completely dependent on the results of the first stage and classifies data using a multilayer perceptron and the Deep Learning technique stochastic gradient descent (SGD). The suggested hybrid approach based on deep learning increases detection rate, accuracy, and lowers false alarms. This model's classification performance was 99.9325, detection rate was 99.9738, and the precision was 0.00093.

A firewall that takes labeled data and configures itself autonomously by writing cautious preventive rules to eliminate false alerts was developed by Haghighi et al. [13]. Z-classifiers employ zero FPR as an administrative criterion, in contrast to ordinary classifiers, which concentrate only on accuracy. A general iterative strategy for accomplishing the goal is then described after it is analytically demonstrated why straightforward modification of existing classifiers, such as SVM, does not produce satisfactory results[14]. The recommended classifier with CART as its core is used to build a firewall for a Power Grid Monitoring System. The KDD CUP'99 dataset is used to test the approach for further evaluation. The results support the effectiveness of the tactic.

A genetic algorithm-based anomaly-based IDS and a novel feature selection method for the Support Vector Machine were proposed by Gharaee et al. [15]. (SVM). The novel model coupled a feature selection method based on genetics with a fitness function innovation to minimise data dimension, boost true positive detection, and decrease false positive detection all at once. Additionally, a significant reduction in training calculation time would occur. Although it had previously been accomplished in independent research, the results demonstrate that the suggested strategy can achieve high accuracy and a low false-positive rate (FPR) at the same time. The GF-SVM model's findings in KDD CUP 99 increased detection accuracy to 99.05 percent for regular traffic, 99.95 percent for DOS class, 99.06 percent for PROBE class, 98.25 percent for R2L, and 100% for U2R.

SVELTE, a model intrusion detection method for the IoT, is recommended in [16]. Implementation and assessment largely target routing attacks such as spoofing or altered information, sinkholes, and selective-forwarding. The technique can be used to identify different kinds of attacks, though. The Contiki OS has SVELTE installed and has undergone rigorous testing. As per the assessment, SVELTE finds all hostile nodes attempts to instigate sinkhole or selective forwarding assaults in the simulated circumstances. However, some false rates happened when malicious nodes were found, therefore the genuine positive rate is not 100%.

To prevent zero-day attacks, Sharma et al. [17] created a specification-based technique for IoT intrusion detection. The recommended IDS uses a specification-based methodology to increase detection, precision, and efficacy (memory and communication overhead) while also offering protection from previously undiscovered assaults. Although the author gives a full explanation of their methodology and analyses in terms of accuracy rate, false-negative rate, and false-positive rate, they do not offer a measuring system to judge the efficacy of the proposed IDS.

To protect smart grid systems from Cyberattacks, a distributed IDS and hierarchical (SGDIDS) is presented in [18]. The designed scheme utilizes classification algorithms like support vector machine (SVM) and artificial immune system to identify the existence, form, and source of an assault in the communicative system (AIS). The SGDIDS can be segregated into two tiers: Edge routers & IoT devices, similar to a regular IoT network. Because of the attack surface for smart grids is fairly similar to that of a conventional IoT network, the study's findings are also applicable to generic IDS for IoT.

Fog computing has been used to create an intrusion detection system [19]. To train a recurrent neural network, the authors updated the back propagation method. The results of the experiment demonstrated the use of adaptive cascaded filtering in conjunction with neural networks' recursive structure, being adaptively altered to various hyper-parameters to maximize the identification of particular sorts of invasions.

III. METHODOLOGY

The proposed approach aims to mitigate two major problems: The first step should be to create a pattern-trained classifier that can successfully separate attack patterns from regular patterns while attaining a lower false-positive rate. The choice of the dataset that covers a diversity of assault types should be taken into account as another problem. The aforementioned problems are addressed using an upgraded Random Forest method.

3.1 Random Forest Classifier

According to Breiman, random forests are collections of trees where each tree is built using a sample taken from the initial training set. To find a new item from each tree in the forest, the input vector will be placed. Each tree makes a vote to specify its preference for the object's class. Out of all the trees in the forest, the forest chooses the category with the highest votes[20]. The formation of forest trees is demonstrated in Figure 1.

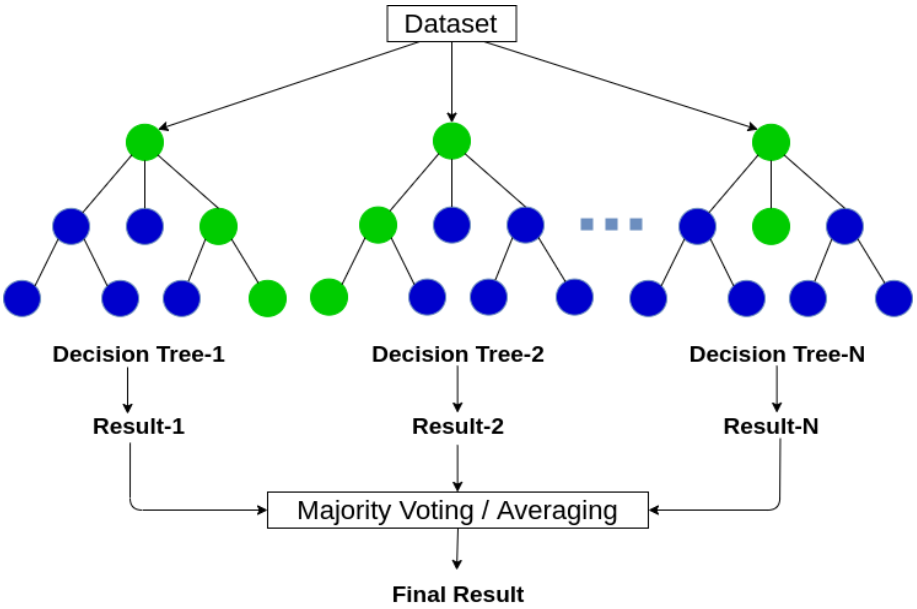

Figure 1: Depiction of Random Forest

1. Assume that the first training data contains *TN* occurrences. Make a bootstrap sample of size *TN* using the initial training data. A fresh tree training dataset will be created using this sample. Data from the native training set except the bootstrap sample is referred to as out-of-bag data.

2. Assume that *F* represents the overall input features in the training set. This bootstrap sample data, where $f < F$, randomly selects only m characteristics for each tree. The characteristics from this collection produce the optimal splits at every node in the tree. The value of *m* ought to remain same as the forest expands.

The error rate of a forest is determined by the strength of its individual trees as well as by the relationships between those trees [21]. Maximizing the accuracy of individual trees reduces the forest error rate while increasing the connectivity raises it. Both the strength and the correlation are impacted by the value of *m*. Therefore, decreasing f decreases the correlation as well as the strength [20].

According to the research, Random Forest (RF) techniques beat single decision tree algorithms in terms of accuracy and huge datasets. Without over-fitting, RF can handle nominal data. Based on the predictions made by the ensemble of trees, test data categorization is selected by a majority of votes [22].

3.2 Proposed Method

Data pre-processing and the RF Classification, as described in Algorithm 1, are the two steps of the proposed approach. Data traffic records are pre-processed in the traffic processing unit to produce data traffic in a format suitable for processing by the RF classifier, and these records are then classified by the smart IDS as an attack (1) or normal (0). The normal/attack classification prediction skills of the proposed model are enhanced using an RF method. The Random Forest is the most significant classification-based traffic analysis engine. In order to detect intrusions, the RF tracks the network traffic patterns leading to the IoT device/system and pops an alert message.

- ***Data Traffic Intensity Processing***

The NSL-KDD dataset is utilised for the model's training, testing, and validation. Data attributes that specify the traffic input to networking systems are inherently erratic. Pre-processing of data traffic is therefore a crucial doorway for the classifier. The pre-processor engine completes three main tasks for the traffic flows on raw traffic data: symbolic to numeric conversion, reduction of features, and data traffic oversampling.

- ***Symbolic-to-Numeric Conversion***

As per the offline traffic data samples, three symbolic fields that come after the starting numeric field recognize the service, protocol, and flag components of connection records. In this step, the attributes from the NSL-KDD dataset are transformed into numerical values. Protocol: UDP = 1, TCP = 2, ICMP = 3 is used to denote the features of a protocol. Service characteristics include Private (16, 16, 30, Netsat) and Flag (pstr, 4,.....,S2= 14), respectively. The attribute weight, which corresponds to the attribute value, was chosen depending on the recurrence of the feature. As the recurrence maximises, the linked numeric number minimises. As a result, the importance of the features with the lowest frequency won't be overshadowed by those with the highest frequency.

In the final stage of dataset codification, the various identities of assault sub-classes are enfolded and encoded to their basic classes. All training dataset tuples are separated into regular and assault types using two enclosure types (normal and attack categorization) for the binary classifier. Four categories—probe, DoS, privilege, and access are used to group the 40 attack labels (classes).

(i) Feature Reduction

In feature reduction, all records in the traffic pattern that have constant valued features are removed from them since they do not influence the tangible outcome of the RF. The different features with zero values were removed from our suggested method, which reduced the number of features from 40 to 25.

(ii) Data traffic Oversampling

Traffic Data oversampling is a prominent phase in the pre-processing of data process to reduce the issue of dataset imbalance. About 125,000 records make up the NSL-KDD dataset. Pure calculations get the following percentages: 67,340, 45,930, 11,654, 992, and 57, respectively, for normal, DoS, probe, privilege, and access records. 50% of the dataset is made up of normal records, followed by DoS and Probe records, with the other classes making up 5% less data from the training data. Our RF displays skewed classification behaviour toward DoS and normal records as a result of this unintentional imbalance, as well as a subpar classification reply towards other assault types that are less frequent. The RF will view privilege and access assaults as noisy signals due to their low frequency resulting in an obstruction in detection.

(iii) Intrusion Detection

Figure 2 depicts the proposed Intrusion Detection framework, and Algorithm 1 demonstrates how it functions. The data has undergone three stages of pre-processing in the data traffic processing engine. The NSL-KDD dataset's symbolic fields were codified into numeric fields as the initial phase in the symbolic-to-numeric conversion process. Next, feature reduction is carried out to eliminate those features with constant values that have no bearing on the outcomes, and last, the most crucial data oversampling is carried out to produce a steady-state of the dataset. The pre-processed data is divided into training and testing data sets after processing. The data is fed to the RF classifier, which performs the classification, and includes 0.8percent records in the training set and 0.2 percent records in the testing set. Our IDE pops a message when the classified data is noticed as an incursion. A Random Forest-based classifier serves as the foundation of the proposed detection engine. A classification or regression ensemble method is known as a Random Forest.

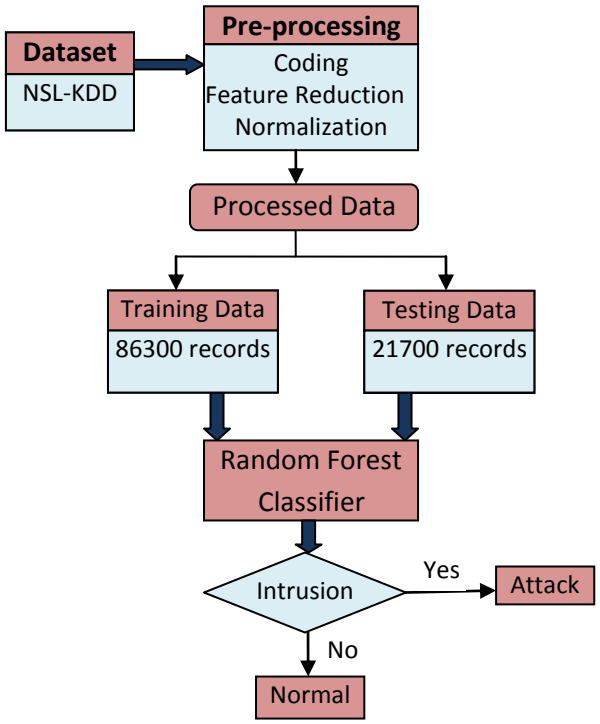

Figure 2: Proposed Intrusion Detection Framework

Algorithm 1: Intrusion Detection System using proposed Random Forest algorithm

I/P: NSL-KDD Dataset
Initialize: RF ( )
O/P: Assault or Normal
1: **Start**
2: data ← Read_Dataset(DS)
3: data ← Feature_Reduction(data)
4: data ← Normalize(data)
5: Train_set, Test_set ← Train_Test_Slit (data,82.45,17.55)
6: RF(Train_set)
7: Status ← RF(Test_set)
8: **if** rank ==0 **then**
9: alert ("Normal Data")
10: **else**
11: alert ("Assault Detected")
12: **end if**
13: **End**

IV. RESULTS AND ANALYSIS

An appropriate technique must offer an accurate classification with a low error rate. The various performance indicators used in our proposed algorithm are discussed subsequently.

$$Accuracy\ (ACC) = TP+TN\ /\ TP+TN+FP+FN \quad (1)$$
$$True\ Positive\ Rate\ (TPR) = TP\ /\ TP+FN \quad (2)$$
$$False\ Positive\ Rate\ (FPR) = FP\ /\ FP+TN \quad (3)$$
$$Precision\ (PRE) = TP\ /\ TP+FP \quad (4)$$
$$F1\_Measure = 2*TPR*PRE/PRE+RECALL \quad (5)$$
$$False\ Negative\ Rate\ (FNR) = FN/\ FN+TP \quad (6)$$
$$Mathew\ Correlation\ Coefficient\ (MCC) = TP*TN - FP*FN/\sqrt{(Tp*FP)(TP+FN)(TN+FP)(TN+FN)} \quad (7)$$

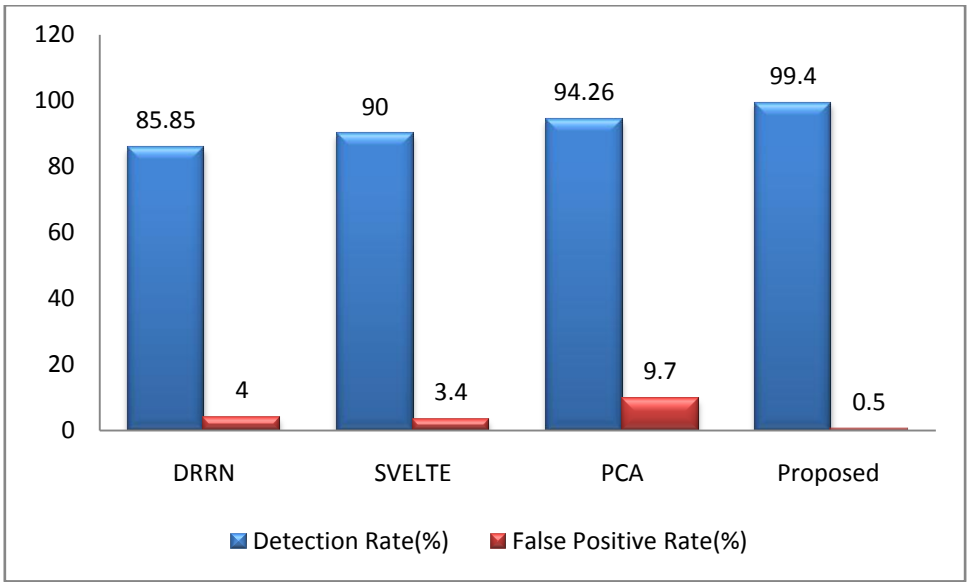

Figure 3: Comparison of the proposed method with other techniques

We concentrated our analysis on studies that specifically addressed IoT security by employing the NSL-KDD dataset. As shown in Table 6, when compared to DRRN[19], SVELTE[16], and GF-SVM[15], our model performed better in terms of FPR and had noticeably better results in the context of True Positive Rate, Accuracy, False Positive Rate, and False Negative Rate. As shown by K and MCC coefficients of 99.54 and 97.60 percent, respectively, our proposed technique achieved significantly good results, guaranteeing the model's stability against low-frequency attacks. Figure 3 compares the false positive rate and true positive rate of the suggested method with those of other techniques.

V. CONCLUSION

IoT networks are vulnerable to attacks, and early identification of illicit activity may prevent the theft of data. Although many IDS have been described in the literature, they are mostly applicable to prevalent networking paradigms, and there has been little research done to develop ML-based IDS for IoT applications. Additionally, we were unable to locate any studies that examined how classifier performance affected IoT-based intrusion detection. Additionally, no prior study has shown how to develop a classifier on IoT hardware. As a result, we concentrated primarily on developing IDS for IoT defense utilizing ML classification techniques. The classifier's accuracy, specificity, sensitivity, and FPR are all assessed. The classifier is benchmarked using the NSL-KDD datasets. The results show that in terms of ACC, True positive rate, FPR, precision, recall, and f1 measure, the proposed approach outperforms existing models. The NSL-KDD dataset has been used as the industry standard for decades, although it is outdated and neither accurately nor sufficiently captures the current network traffic. Therefore, it will be necessary to assess the models either in offline mode on the latest dataset that has all the properties of the original network or in a realistic network application in the future.

REFERENCES

[1]. J. Pacheco and S. Hariri, "Anomaly behavior analysis for IoT sensors," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 4, p. e3188, Apr. 2018.
[2]. B. Rana, "A systematic survey on internet of things : Energy efficiency and interoperability perspective," no. August, pp. 1–41, 2020.
[3]. Y. Meidan *et al.*, "N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul. 2018.
[4]. B. Rana, Y. Singh, and H. Singh, "Metaheuristic Routing : A Taxonomy and Energy-Efficient Framework for Internet of Things," *IEEE Access*, vol. 9, pp. 155673–155698, 2021.
[5]. S. Panchiwala and M. Shah, "A Comprehensive Study on Critical Security Issues and Challenges of the IoT World," *J. Data, Inf. Manag.*, vol. 2, no. 4, pp. 257–278, Dec. 2020.
[6]. B. Rana and Y. Singh, "Internet of Things and UAV: An Interoperability Perspective," *Unmanned Aer. Veh. Internet Things*, pp. 105–127, 2021.
[7]. B. Rana and S. Yashwant, "Duty-Cycling Techniques in IoT: Energy-Efficiency Perspective," in *International Conference on Recent Innovations in Computing(ICRIC-2021)*, 2021.
[8]. G. Hameed, Y. Singh, S. Haq, and B. Rana, "Blockchain-Based Model for Secure IoT Communication in Smart Healthcare," pp. 715–730, 2022.
[9]. H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K. K. R. Choo, "A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks," *IEEE Trans. Emerg. Top. Comput.*, vol. 7, no. 2, pp. 314–323, 2019.
[10]. P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things,"

Int. Conf. Wirel. Mob. Comput. Netw. Commun., pp. 600–607, 2013.
[11]. D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *Int. J. Inf. Manage.*, vol. 49, pp. 533–545, Dec. 2019.
[12]. M. Tabash, M. A. Allah, and B. Tawfik, "Intrusion Detection Model Using Naive Bayes and Deep Learning Technique," *Int. Arab J. Inf. Technol.*, vol. 17, no. 2, 2020.
[13]. M. Sayad Haghighi, F. Farivar, and A. Jolfaei, "A Machine Learning-based Approach to Build Zero False-Positive IPSs for Industrial IoT and CPS with a Case Study on Power Grids Security," *IEEE Trans. Ind. Appl.*, pp. 1–1, Jul. 2020.
[14]. H. Sedjelmaci, S. M. Senouci, and T. Taleb, "An accurate security game for low-resource iot devices," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9381–9393, Oct. 2017.
[15]. H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," *2016 8th Int. Symp. Telecommun. IST 2016*, pp. 139–144, Mar. 2017.
[16]. S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.
[17]. V. Sharma, I. You, K. Yim, I. R. Chen, and J. H. Cho, "Briot: Behavior rule specification-based misbehavior detection for IoT-embedded cyber-physical systems," *IEEE Access*, vol. 7, pp. 1–25, 2019.
[18]. Y. Zhang, L. Wang, W. Sun, R. C. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.
[19]. M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simul. Model. Pract. Theory*, vol. 101, p. 102031, May 2020.
[20]. A. N. Iman and T. Ahmad, "Improving Intrusion Detection System by Estimating Parameters of Random Forest in Boruta," *Proceeding - ICoSTA 2020 2020 Int. Conf. Smart Technol. Appl. Empower. Ind. IoT by Implement. Green Technol. Sustain. Dev.*, Feb. 2020.
[21]. P. Goyal, A. K. Sahoo, and T. K. Sharma, "Internet of things: Architecture and enabling technologies," *Mater. Today Proc.*, vol. 34, pp. 719–735, Jan. 2021.
[22]. R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," *Proc. 2017 Int. Conf. Data Softw. Eng. ICoDSE 2017*, vol. 2018-January, pp. 1–6, Feb. 2018.