Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 1003–1013

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Anomaly detection using ensemble techniques for boosting the security of intrusion detection system

Owais Bukhari[a], Parul Agarwal[a,*], Deepika Koundal[b], Sherin Zafar[a]

[a]*Jamia Hamdard, Hamdard Nagar, New Delhi-110062, India*
[b]*University of Petroleum & Energy Studies, Dehradun, India*

Abstract

IoT-based applications have witnessed a rapid surge in deployment in various domains. IoT infrastructure is the nervous system responsible for the effective functioning of Smart Cities. Nevertheless, the full-fledged deployment of IoT-based applications exposes this infrastructure to a high risk of cyberattacks. Since IoT devices establish communication with cloud services via inbuilt sensors, the probability of sabotaging the communication channel by malicious attacks is always high. This paper aims to explore an anomaly detection method that makes use of techniques like Support Vector Machine (SVM), Artificial Neural Networks (ANN), k- Nearest Neighbor (KNN), Linear Regression (LR), Decision Trees (DT), and Random Forest (RF) to neutralize threats and boost the cybersecurity of a smart city. The paper goes on to examine the role of ensemble techniques like bagging and boosting to provide an additional security layer to the detection architecture. This is where the paper departs from the erstwhile approaches that have revolved around single classifiers to boost the detection system and have not considered the integration of cross-validation and feature selection. The experimental results conducted on the datasets UNSW-BC15 and CICIDS2017 and several measures like Accuracy, Precision, Recall, and F1 Score establish that the proposed approach outperforms various state-of-the-art methods used in detecting rare attacks.

Keywords: IoT, smart city, SVM, decision tree, KNN, linear Regression, ANN, cybersecurity, bagging, boosting, intrusion detection system, ensemble techniques.

1. Introduction

Here The Internet of things is the seamless integration of various devices like chatbots, medical devices, smart robots, humanoids, and other smart devices to promote effective communication of information between them. As per

10.1016/j.procs.2023.01.080

a report by Statista, the total number of devices powered by IoT will touch 50 Billion by 2025. This number is all set to cross the hundred billion mark by 2030 [1-2]. But, several gaps of vulnerability in our smart systems expose them to various kinds of cyberattacks [3-4]. Different types of IoT devices and applications run in the contours of a smart city. In [5], the authors propose IoT and machine learning-based analytical approaches for smart cities. As such, even the smallest of the cyber-attack can get access to the very personal details of a citizen without his prior knowledge which may lead to a snowball effect and disintegrate the entire security infrastructure [6]. One of the fundamental questions that is pending before the cybersecurity analysts are to secure the security infrastructure and the network in such a manner that the various types of attacks are disrupted before they halt the orthodox functioning of IoT devices [7]. In the present environments of a smart city, the cloud environment provides a wide range of storage, processing, and computational capabilities. As such, the streak of cloud migration has taken place that has led to the problem of increased latency, congestion, and attacks [8]. Two prime solutions that have been forwa3rded to counter these problems include edge computing and fog computing [9]. While the fog layer is equipped with advanced computational capabilities, the edge layer has a very low latency rate as it processes data close to the source where data is generated [10]. Whenever computations are carried out in the fog layer or near the edge, less interruption of the ongoing computational processes and large-scale damage to the underlying infrastructure will be presented at the first instance [11].

The major contributions of this paper are:

Perform related- studies that have used ML and ensemble techniques for mitigating cybersecurity threats.

To propose an intrusion detection scheme based upon ML and its techniques for the analysis of traffic in fog networks that circum-ambient the IoT infrastructure.

To establish using results the instrumental precision that ensemble modeling techniques attain in threat identification as compared to single classifiers.

The major findings of the work are:

After the experimental results conducted on the datasets UNSW-BC15 and CICIDS2017, we took a note of several measures like Accuracy, Precision, Recall, and F1 Score. We established that the proposed approach outperforms the previous methods used in detecting rare cyber attacks in vulnerable environments.

2. Related Work

Ensemble techniques have found their applications as discussed in [12-13]. In this section, we attempt to bring out the most prominent studies that have harnessed machine learning and ensemble techniques to mitigate cybersecurity threats.

2.1 *The Overlap between Intrusion Detection and Machine Learning*

In [14], a methodology was introduced to predict the behavior of IoT systems with the aid of machine learning techniques, and this was done by observing the transmission of information between devices in a distributed multidimensional microservice environment. In this environment, the microservice model that was conceived made use of k-means clustering and BIRCH-based techniques. In [15,16], an industrial IoT site was considered with the aim to detect attacks that were executed via malicious network nodes. The methodology that was used in this research was TLPD or Trust Light probe-based defense. To detect both onsite and offsite attacks, they designed a framework that could identify different kinds of anomalies using this probe and also carried out measurements related to confidence estimation. The model achieved an accuracy of about 95.4 percent. In [17], classification technique along with dimensionality reduction was used in IoT backbone networks. This type of model was able to detect very low-frequency attacks (U2R and R2L) from the NSL-KDD data set. Linear discriminant analysis and principal component analysis were used to extract different types of features from a data set which was then subjected to a new base algorithm and K nearest-neighbor techniques. The model achieved an accuracy of about 87.8 % in the detection of various kinds of anomalies. In [18,19], the methodology of Extreme ML was used for detecting attacks in the cloud ecosystem. This type of architecture powered by Extreme ML was effectively used for the computation of various clusters of data outsourced from fog computing. The accuracy achieved by this study was about 94.5 percent.

2.2 *The Overlap between Intrusion Detection System and Ensemble Techniques*

In [20], several ML techniques have been used for analysis, and comparison is drawn so that the accuracy is improved over the base classifier. In the ensemble technique proposed by them, they made use of the Gain ratio feature selection technique. They evaluated the performance of NSL-KDD data sets and achieved an accuracy of about 96.02%. Table 1 summarizes a few selected works that detect cyber-attacks in smart cities using various Machine Learning techniques.

Table 1. A summary of selected works for detecting Cyber Attacks in Smart Cities primarily dominated by IoT infrastructure.

Reference	Dataset	Year	Technique used	Category	Accuracy
[14]	NSL-KDD	2018	k-means clustering	Binary	95.6
[15]	Own	2018	TLPD	Multi-class	95.4
[16]	Own	2018	Ensemble	Multi-class	87.8
[17]	NSL-KDD	2019	k-nearest neighbor	Multi-class	94.5
[19]	KDD-99	2019	Extreme ML	Binary	94.03
[20]	NSL-KDD	2019	Random Forest, SVM	Multi-class	96.02
[21]	CIC-IDS2017	2020	CFS	Multi-class	98.8

In [21, 22], the bootstrapping ensemble technique was used and the model achieved an accuracy of 88.67 percent while relying on an NSL-KDD dataset. In [23], a bat algorithm and feature selection based upon correlation (CFS) is used. In addition to this, Forest by penalizing attributes algorithms or PFA was also used in this study and the datasets used consisted of CIC-IDS2017. The results showed an efficiency of 98.8 percent. In [24], the researchers aim to detect zero-day attacks by making use of a dataset that owes its origin to a fully functional IoT ecosystem with network traffic in full flow. Consequently, this type of ecosystem proves to be a vulnerable hotbed for cyber-attacks. As such, this provides a rich data resource that can be leveraged to examine multiple intrusion detection systems. In [25], the usage of a hybrid intrusion detection system is commendable but the concern is that it may, in some capacity, compromise the classifier and even class SVM. While the results are compared to the Signature Intrusion Detection System, a general query that arises in the mind of the reader is the choice of selection of the hybrid system to this particular method only. It is highly probable that the extension of the comparison to Anomaly based Intrusion Detection System (AIDS) may have yielded different results. In [26], the usage of random forest models and decision trees as base classifiers along with bagging and boosting ensemble methods is done. The dataset used is NSL-KDD on the basis of which major observations are done. The final results confirm that bagging along with decision trees yields higher accuracy while intrusion detection. Different datasets have been used in varied contexts in [27] but the prime motive has been the same, that is, to use machine learning algorithms for improving the accuracy of intrusion detection systems. While the choice of dataset in other studies may be driven by various factors, we find that not many of the earlier studies lay impetus on concurrency, a concern which stands addressed in our work. We don't limit

ourselves to identifying the anomalous state of data alone but move on to zero in on the exact type of attacks that occur in fog nodes.

3. Proposed work

The architectural framework of an IoT-based smart city is described in the form of three interconnected layers in the next section.

3.1 *Architectural Framework*

In the working environs of a smart city, we witness the integration of technologies like IoT and other smart systems which not only lead to an effective exchange of information but also help in the maintenance of other services. The different technologies operating in a smart city help in the advancement of domains like healthcare, education, logistics, pollution control, and energy consumption [28]. Generally speaking, the architectural framework of a smart city comprises three layers: the terminal layer, the fog layer, and the cloud layer. The storage resources that include service and other kinds of machines are contained in the cloud layer and they help in the maintenance and processing of voluminous data [29]. The fog layer is the connecting layer between the terminal layer and the cloud layer. The terminal layer communicates with various types of devices and helps in channelizing information exchange between IoT devices and the sensors and collects structured and unstructured data [30]. Figure 1describes the architectural framework of an IoT-based smart city.

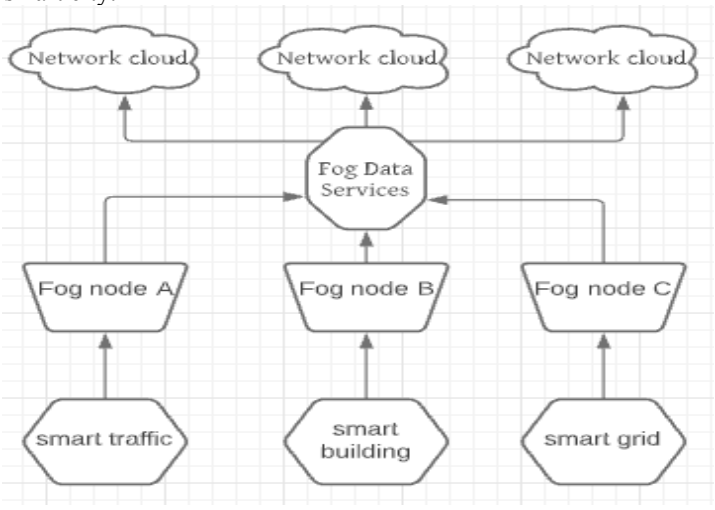


Figure 1**:** The architectural framework of an IoT-based smart city

3.2 *Working Model*

An outline of the working model is demonstrated in Fig. 2. The model works by tracking the traffic generated across the network and sending it across each fog node. These nodes are placed very close to the IoT sensors [31]. This means that the identification of cyberattacks in the vicinity of these nodes will become relatively easy in comparison to the detection of such cyber-attacks near the cloud center. This will not only help in the quick detection of attacks and prevention of the disruption of services in the IoT infrastructure but will also notify the network administrators of the possibility of such attacks as and when they occur in near future. This will also allow the network administrators to upgrade their system and fill the voids in their security shields [32]. The IDS that we are talking about can primarily be classified into two categories: host-based and Network-based intrusion detection systems. We chose to work closely with the network intrusion detection system. The reason why we avoid a host-based syste is that it requires the software to be installed on each device for tracking malicious activities [33]. So, working on each device individually may not be suitable while trying to conceive a generic security framework for the entire Smart City.

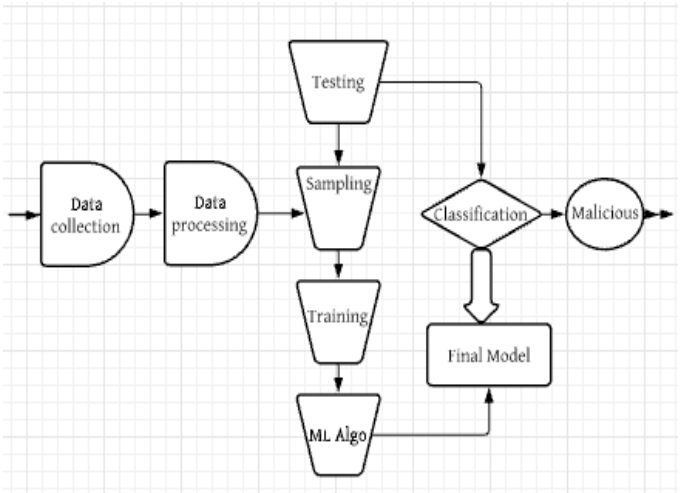


Fig 2. Working Model for the Intrusion Detection System

3.2.1 *Datasets*

The study revolved around the usage of two datasets viz UNSW-NB15 [34] and CICIDS2017 [35]. There are two main reasons for using these datasets. Their relevance to various concepts in this paper-like smart city infrastructure is especially appealing. They have been used as a precursor in various studies related to attacks in an IoT-based setup. The first dataset consists of a sample space of more than 2 million entries. A random sample consisting of 164231 entries was selected. The training set consisted of 139161 entries and the test dataset consisted of 25,070 entries. The second dataset has a sample space of 2720634 entries and the total number of features is 77.

3.2.2 *Feature selection*

Selecting only those features that can effectively cater to the various requirements of our model so that overfitting is reduced and accuracy is improved is essential. To select the features, a threshold limit was decided for both datasets using information gain as the basis for the same. Information gain beyond a certain limit is always a precursor to determining cyberattacks with a higher degree of accuracy. The threshold limit for the information gain of the first dataset was fixed at 0.4 and the corresponding limit for the second dataset was fixed at 0.7. Accordingly, out of 40 and 80 features corresponding to the first and second datasets, 25 features were considered that were common to both datasets. The information gain ratio is displayed in the adjoining figure for the two datasets in Table 2 and Table 3.

Table 2. Information gain ratio for various features related to the dataset UNSW-NB15

Feature number	Feature name	Ratio	Feature number	Feature name	Ratio
35	dload	0.767	30	service	0.675
10	porto	0.656	2	sttl	0.766
1	dtll	0.986	31	rate	0.543
34	dur	0.4554	34	ct_state_tti	0.786
27	dmean	0.876	8	smean	0.654
11	sload	0.657	7	dbytes	0.865
6	sbytes	0.898	26	smean	0.786

Table 3. Information gain ratio for various features relating to the ICIDS2017 dataset

Feature number	Feature name	Ratio	Feature number	Feature name	Ratio
52	Packet Size	1.675	40	Packet Length mean	1.654
41	Packet Length	1.654	42	Variance	1.546
18	Destination port	1.987	1	Flow duration	1.765
36	Segment size	1.265	14	Flow bytes	1.876
15	Fwd IAT Max	1.867	16	Flow IAT Mean	1.435

3.2.3 *Model Building and Performance Evaluation*

The Machine learning techniques that were utilized for model building and performance evaluation include RF, SVM, KNN, and ANN. For designing the architecture of the intrusion detection system, machine learning was combined with various types of base models so that one optimal predictive model could be obtained [36]. The aggregate of these models was considered and the ensemble was used to combine all such models into a consolidated one. The principle operating behind the final model was that the aggregate of the models put together forms a stronger model which increases the overall accuracy [37].
Parameters like accuracy, precision, recall (True Positive Rate), False Positive Rate, the ROC curve and F1-score, were used for performance evaluation for detecting of anomalies in IoT applications [38]. The parameters are described below:

$$Accuracy = (True\ Positive + True\ Negative) \div (Total\ Positive + Total\ Negative) - - - -(1)$$

$$Recall\ (TPR) = True\ Positive \div (True\ Positive + False\ Negative) - - - - - - - - - -(2)$$

$$FPR\ (False\ positive\ rate) = False\ Positive/(False\ Positive + True\ Negative) - - - - -(3)$$

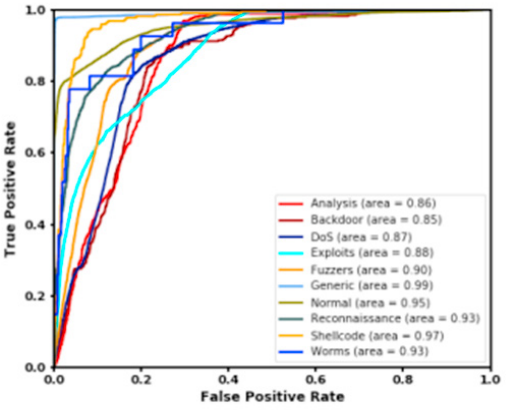
$$F1 - Score = 2 * Precision * Recall \div (Precision + Recall) - - - - - - - - - - - - - - - (4)$$

4. **Experimental settings**

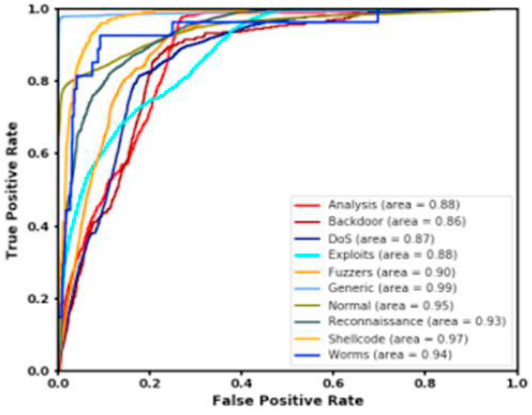
For the experimental settings, we used HP notebook with specifications-(Processor: 8th Generation hexa-core Intel Core i7 processor.Graphics. Integrated: Intel UHD Graphics 630.Memory: 16GB DDR4-2666 SDRAM.Storage: 512GB PCIe® NVMe™ M.2 SSD). We used the python programming language with frequent use of libraries like Matplotlib and Pandas. We operated the base classifier and the ensemble classifier for an initial performance check by randomly dividing the data set into ten subsets(nine for model classifier and one as a test set).

5. **Results**

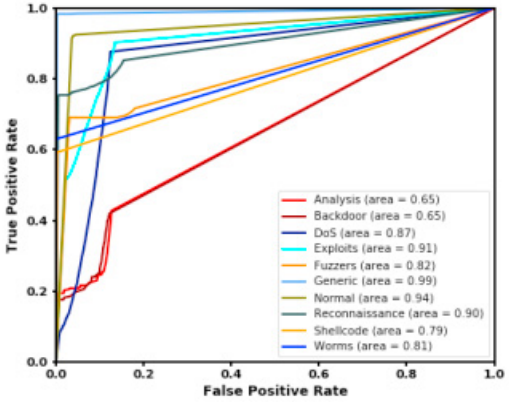
The receiver operating characteristic (ROC) curve describes the performance of the classifier and the various decision thresholds in the plot. Figure 3(a-h) depicts the ROC curve. For testing the performance of the model, a 10-fold cross-validation technique was used. The data sets were segregated into a sample space of 10. In this sample space, the last one was used as a test set while 9 others were used to construct the model. This procedure was repeated and the mean accuracy was obtained from each classifier. An unclassified sample was taken and was categorized into one of the ten samples for the first dataset and one of the eight samples for the second dataset.


True Positive Rate
False Positive Rate
Analysis (area = 0.86)
Backdoor (area = 0.85)
DoS (area = 0.87)
Exploits (area = 0.88)
Fuzzers (area = 0.90)
Generic (area = 0.99)
Normal (area = 0.95)
Reconnaissance (area = 0.93)
Shellcode (area = 0.97)
Worms (area = 0.93)

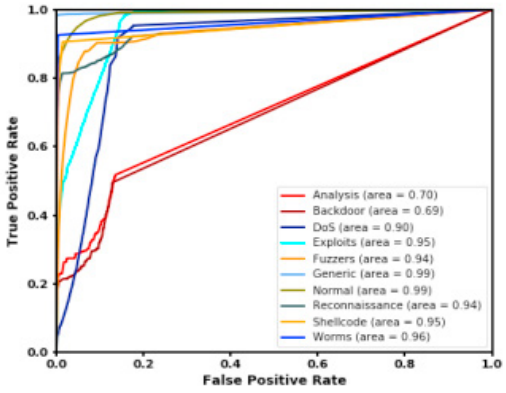
(a)


True Positive Rate
False Positive Rate
Analysis (area = 0.88)
Backdoor (area = 0.86)
DoS (area = 0.87)
Exploits (area = 0.88)
Fuzzers (area = 0.90)
Generic (area = 0.99)
Normal (area = 0.95)
Reconnaissance (area = 0.93)
Shellcode (area = 0.97)
Worms (area = 0.94)

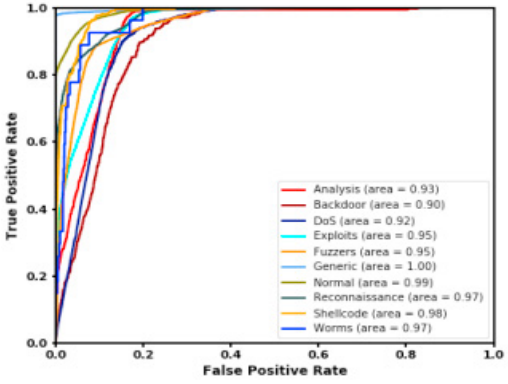
(b)


True Positive Rate
False Positive Rate
Analysis (area = 0.65)
Backdoor (area = 0.65)
DoS (area = 0.87)
Exploits (area = 0.91)
Fuzzers (area = 0.82)
Generic (area = 0.99)
Normal (area = 0.94)
Reconnaissance (area = 0.90)
Shellcode (area = 0.79)
Worms (area = 0.81)

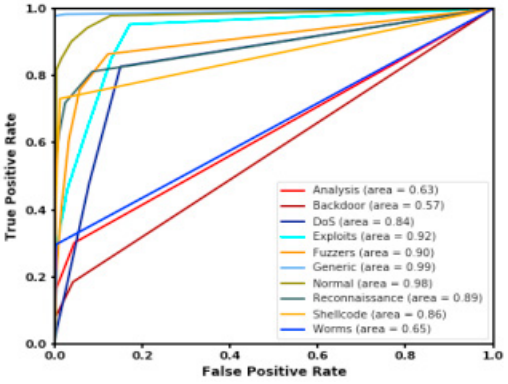
(c)


True Positive Rate
False Positive Rate
Analysis (area = 0.70)
Backdoor (area = 0.69)
DoS (area = 0.90)
Exploits (area = 0.95)
Fuzzers (area = 0.94)
Generic (area = 0.99)
Normal (area = 0.99)
Reconnaissance (area = 0.94)
Shellcode (area = 0.95)
Worms (area = 0.96)

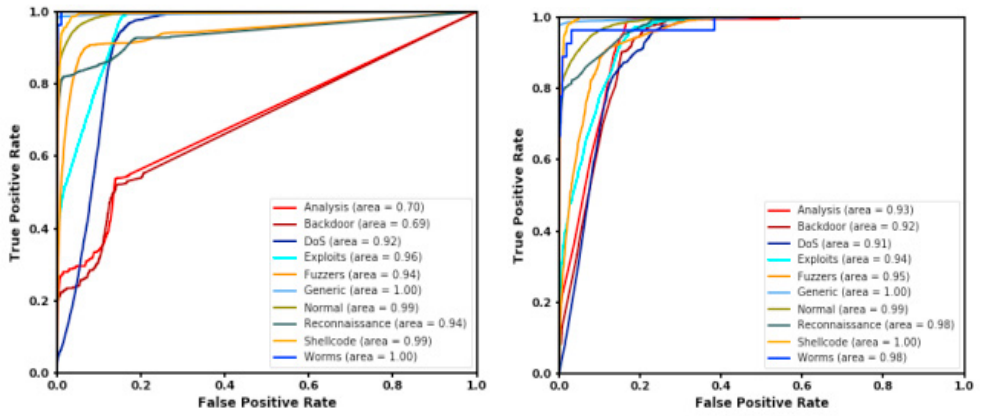
(d)


True Positive Rate
False Positive Rate
Analysis (area = 0.93)
Backdoor (area = 0.90)
DoS (area = 0.92)
Exploits (area = 0.95)
Fuzzers (area = 0.95)
Generic (area = 1.00)
Normal (area = 0.99)
Reconnaissance (area = 0.97)
Shellcode (area = 0.98)
Worms (area = 0.97)

(e)


True Positive Rate
False Positive Rate
Analysis (area = 0.63)
Backdoor (area = 0.57)
DoS (area = 0.84)
Exploits (area = 0.92)
Fuzzers (area = 0.90)
Generic (area = 0.99)
Normal (area = 0.98)
Reconnaissance (area = 0.89)
Shellcode (area = 0.86)
Worms (area = 0.65)

(f)



(g) (h)

Fig. 3. ROC curve for various techniques. (a) Curve for logistic regression. (b) Curve for SVM. (c) Curve for decision tree. (d) Curve for the random forest. (e) Curve for ANN. (f) Curve for KNN. (g) Curve for Bagging ensemble technique. (h) Curve for Boosting ensemble technique.

The accuracy using the support vector machine was found to be 90.50% and it was 70.3 8% for the decision tree. It was 91% for the random forest and 79.5% for the artificial neural network. It was 98.8 percent for the K nearest neighbor. The accuracy obtained for the ensemble method was about 80.25%. It was 98.6% for boosting method and it was about 98.8 percent for stacking. The precision for Linear Regression (LR) was found to be 71%, 89% for Support Vector Machine (SVM), 69% for Decision Tree (DT), 89% for Random Forest (RF), 72% for ANN, and 86.5% for K-Nearest Neighbor (KNN). The precision of ensemble methods was found to be 79% and 91% for bagging, 88% and 97% for boosting, and 86% and 97% for stacking. The recall for LR was found to be 69%, 89% for SVM, 82% for DT, 88% for RF, 94% for ANN, and 97% for KNN. The F1-score was found to be 69% for LM, 89% for SVM, 69% for DT, 88% for RF, 92% for ANN, and 96% for KNN.

Table 4. Model validation: Comparative chart of multi-class classification performance based on the UNSW-NB15 dataset

	Proposed Model			Current models		
Algorithm	TPR	FPR	F1	TPR	FPR	F1
LR	0.71	0.03	0.74	0.87	0.01	0.86
SVM	0.62	0.05	0.78	0.84	0.21	0.56
DT	0.65	0.08	0.87	0.76	0.32	0.67
RF	0.76	0.05	0.98	0.87	0.34	0.87
ANN	0.56	0.06	0.89	0.89	0.32	0.07
KNN	0.78	0.09	0.78	0.78	0.89	0.67
Bagging	0.67	0.07	0.87	0.89	0.99	0.56
Boosting	0.89	0.03	0.77	0.99	0.78	0.87
Stacking	0.99	0.09	0.55	0.67	0.78	0.87

The F1-score of ensemble methods like bagging was found to be 79% and 94%. For boosting, it was found to be 88% and 96% and for stacking, it was found to be 81% and 97%. In the bagging technique, the RF, for boosting, DT, and for stacking RF, is used as the base learner. The best results upon comparing the performance of the algorithms were

obtained for decision trees and random forests. The support vector machine showed relatively poor results. Among the ensemble techniques, the stacking method showed the best performance in comparison to others. This is shown in Table 4 and Table 5. FPR and TPR measure the ability to distinguish between classes. It is known that for good performance, the values of F1 (harmonic mean between Precision and Recall) should be close to 1. As can be seen from the tables, the proposed model exhibits for most of the cases, the value of F1 higher than the current approach.

Table 5. Model Validation: Comparative chart of multi-class classification performance for ICIDS2017 dataset

	Proposed Model			Current Models		
Algorithm	TPR	FPR	F1	TPR	FPR	F1
LR	0.71	0.03	0.74	0.87	0.01	0.86
SVM	0.72	0.06	0.87	0.76	0.44	0.85
DT	0.67	0.07	0.78	0.67	0.43	0.78
RF	0.76	0.04	0.67	0.56	0.56	0.76
ANN	0.77	0.07	0.87	0.76	0.67	0.56
KNN	0.65	0.09	0.67	0.56	0.76	0.54
Bagging	0.76	0.06	0.87	0.65	0.76	0.87
Boosting	0.73	0.08	0.67	0.56	0.87	0.78
Stacking	0.67	0.09	0.87	0.76	0.78	0.67

6. Conclusion

The scope of the ensemble-based learning methodology was extended to identify various kinds of Cyberattacks within the IoT framework of a smart city. The experimental results revealed that the ensemble approach that we adopted yielded better results in comparison to the single models that were earlier employed for identifying the attacks. Any breach of data would not only expose the critical information of citizens but would bring the entire IoT infrastructure under severe threat. Though, Soft computing can play a pivotal role in mitigating IoT-based abuse as discussed in [39]. Our research can serve as a foundational element to block such threats. Moreover, it can generate deep insights for not only preventing severe attacks in the IoT infrastructure but can also pave the way for the development of a future intrusion detection system that is virtually impossible to penetrate.

We conclude that the experimental results we obtained on the basis of the datasets UNSW-BC15 and CICIDS2017 can be used as a major lead for detection of rare attacks in the IoT environs of a smart city. Several parameters used in the work like Accuracy, Precision, Recall, and F1 Score are a quantitative testimony of the superiority of our experimental settings.

References

[1] Lai, Liang-Bin, Ray-I. Chang, and Jen-Shiang Kouh. "Detecting network intrusions using signal processing with query-based sampling filter." EURASIP Journal on Advances in Signal Processing 2009 (2008): 1-8.

[2] Znaidi, Wassim, Marine Minier, and Stéphane Ubéda. "Hierarchical node replication attacks detection in wireless sensor networks." International Journal of Distributed Sensor Networks 9.4 (2013): 745069.

[3] Abeshu, Abebe, and Naveen Chilamkurti. "Deep learning: The frontier for distributed attack detection in fog-to-things computing." IEEE Communications Magazine 56.2 (2018): 169-175.

[4] Chowdhury, Abdullahi, Gour Karmakar, and Joarder Kamruzzaman. "The co-evolution of cloud and IoT applications: Recent and future trends." Handbook of Research on the IoT, Cloud Computing, and Wireless Network Optimization. IGI Global, 2019. 213-234.

[5] Hassan, Syed Imtiyaz, and Parul Agarwal. "Analytical approach to sustainable smart city using IoT and machine learning." Big Data, IoT, and Machine Learning. CRC Press, 2020. 277-294.

[6] Restuccia, Francesco, Salvatore D'Oro, and Tommaso Melodia. "Securing the internet of things: New perspectives and research challenges." arXiv preprint arXiv:1803.05022 (2018).

[7] Yar, Majid, and Kevin F. Steinmetz. Cybercrime and society. Sage, 2019.

[8] Aijaz, Iflah, Sheikh Mohammad Idrees, and Parul Agarwal. "An Empirical Study on Analysing DDoS Attacks in Cloud Environment." Advances in Intelligent Computing and Communication. Springer, Singapore, 2021. 295-305.

[9] Kumar, Akshi, and Abhilasha Sharma. "Ontology driven social big data analytics for fog enabled sentic-social governance." Scalable Computing: Practice and Experience 20.2 (2019): 223-236.

[10] Habibzadeh, Hadi, Tolga Soyata, Burak Kantarci, Azzedine Boukerche, and Cem Kaptan. "Sensing, communication and security planes: A new challenge for a smart city system design." Computer Networks 144 (2018): 163-200.

[11] Guha, Sudipto, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. "Clustering data streams: Theory and practice." IEEE transactions on knowledge and data engineering 15, no. 3 (2003): 515-528.

[12] Kumar, Akshi, Kartik Anand, Simran Jha, and Jayansh Gupta. "Online Credit Card Fraud Analytics Using Machine Learning Techniques." In Data Management, Analytics and Innovation, pp. 107-120. Springer, Singapore, 2021.

[13] Basheer, Shakila, Kapil Kumar Nagwanshi, Surbhi Bhatia, Sipi Dubey, and G. R. Sinha. "FESD: An approach for biometric human footprint matching using fuzzy ensemble learning." IEEE Access 9 (2021): 26641-26663.

[14] Galal, Hisham Shehata, Yousef Bassyouni Mahdy, and Mohammed Ali Atiea. "Behavior-based features model for malware detection." Journal of Computer Virology and Hacking Techniques 12, no. 2 (2016): 59-67.

[15] Hansen, Lars Kai, and Peter Salamon. "Neural network ensembles." IEEE transactions on pattern analysis and machine intelligence 12, no. 10 (1990): 993-1001.

[16] Hossain, Md Mahmud, Maziar Fotouhi, and Ragib Hasan. "Towards an analysis of security issues, challenges, and open problems in the internet of things." In 2015 ieee world congress on services, pp. 21-28. IEEE, 2015.

[17] Pahl, Marc-Oliver, and François-Xavier Aubet. "All eyes on you: Distributed Multi-Dimensional IoT microservice anomaly detection." In 2018 14th International Conference on Network and Service Management (CNSM), pp. 72-80. IEEE, 2018.

[18] Liu, Xiao, Yuxin Liu, Anfeng Liu, and Laurence T. Yang. "Defending ON–OFF attacks using light probing messages in smart sensors for industrial communication systems." IEEE Transactions on Industrial Informatics 14, no. 9 (2018): 3801-3811.

[19] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In 2009 IEEE symposium on computational intelligence for security and defense applications, pp. 1-6. IEEE, 2009.

[20] Pajouh, Hamed Haddad, Reza Javidan, Raouf Khayami, Ali Dehghantanha, and Kim-Kwang Raymond Choo. "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks." IEEE Transactions on Emerging Topics in Computing 7, no. 2 (2016): 314-323.

[21] Alrashdi, Ibrahim, Ali Alqazzaz, Esam Aloufi, Raed Alharthi, Mohamed Zohdy, and Hua Ming. "Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning." In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0305-0310. IEEE, 2019.

[22] Almomani,Iman, Bassam Al-Kasasbeh,2 and Mousa AL-Akhras. "WSN-ds: A dataset for intrusion detection systems in wireless sensor networks," Journal of Sensors, pp 1-17, 2016.

[23] Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." ICISSp 1 (2018): 108-116.

[24] Taneja, Mohit, and Alan Davy. "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm." In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 1222-1228. IEEE, 2017.

[25] Illy, Poulmanogo, Georges Kaddoum, Christian Miranda Moreira, Kuljeet Kaur, and Sahil Garg. "Securing fog-to-things environment using intrusion detection system based on ensemble learning." In 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-7. IEEE, 2019.

[26] Shanthamallu, Uday Shankar, Andreas Spanias, Cihan Tepedelenlioglu, and Mike Stanley. "A brief survey of machine learning methods and their sensor and IoT applications." In 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1-8. IEEE, 2017.

[27] Moustafa, Nour. Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic. Diss. University of New South Wales, Canberra, Australia, 2017.

[28] Khoda, M.E., Imam, T., Kamruzzaman, J., Gondal, I. and Rahman, A., 2019. Robust malware defense in industrial IoT applications using machine learning with selective adversarial samples. IEEE Transactions on Industry Applications, 56(4), pp.4415-4424.

[29] Cugurullo, Federico. "Exposing smart cities and eco-cities: Frankenstein urbanism and the sustainability challenges of the experimental city." Environment and Planning A: Economy and Space 50, no. 1 (2018): 73-92.

[30] Ismagilova, Elvira, Laurie Hughes, Nripendra P. Rana, and Yogesh K. Dwivedi. "Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework." Information Systems Frontiers (2020): 1-22.

[31]Huda, Shamsul, Jemal Abawajy, Mamoun Alazab, Mali Abdollalihian, Rafiqul Islam, and John Yearwood. "Hybrids of support vector machine wrapper and filter based framework for malware detection." Future Generation Computer Systems 55 (2016): 376-390.

[32] Gaikwad, D. P., and Ravindra C. Thool. "Intrusion detection system using bagging with partial decision treebase classifier." Procedia Computer Science 49 (2015): 92-98.

[33] Jabbar, M. Akhil, Rajanikanth Aluvalu, and S. Sai Satyanarayana Reddy. "Cluster based ensemble classification for intrusion detection system." In Proceedings of the 9th International Conference on Machine Learning and Computing, pp. 253-257. 2017.

[34] https://research.unsw.edu.au/projects/unsw-nb15-dataset

[35] https://www.unb.ca/cic/datasets/ids-2017.html

[36] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." Information Security Journal: A Global Perspective 25, no. 1-3 (2016): 18-31.

[37] Tawalbeh, Mais, Muhannad Quwaider, and A. Tawalbeh Lo'ai. "Authorization model for IoT healthcare systems: case study." In 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 337-342. IEEE, 2020.

[38] Anitha Avula, V., and Arba Asha. "Improving Prediction Accuracy Using Hybrid Machine Learning Algorithm on Medical Datasets," International Journal of Scientific & Engineering Research, vol. 9, pp 1461-1467, 2018.

[39] He, Hongmei, Carsten Maple, Tim Watson, Ashutosh Tiwari, Jörn Mehnen, Yaochu Jin, and Bogdan Gabrys. "The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence." In 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 1015-1021. IEEE, 2016.