Machine Learning-Driven Intrusion Detection for Contiki-NG-Based IoT Networks Exposed to NSL-KDD Dataset

Jinxin Liu
jliu367@uottawa.ca
University of Ottawa
Ottawa, ON

Burak Kantarci
burak.kantarci@uottawa.ca
University of Ottawa
Ottawa, ON

Carlisle Adams
cadams@uottawa.ca
University of Ottawa
Ottawa, ON

ABSTRACT

Wide adoption of Internet of Things (IoT) devices and applications encounters security vulnerabilities as roadblocks. The heterogeneous nature of IoT systems prevents common benchmarks, such as the NSL-KDD dataset, from being used to test and verify the performance of different Network Intrusion Detection Systems (NIDS). In order to bridge this gap, in this paper, we examine specific attacks in the NSL-KDD dataset that can impact sensor nodes and networks in IoT settings. Furthermore, in order to detect the introduced attacks, we study eleven machine learning algorithms and report the results. Through numerical analysis, we show that tree-based methods and ensemble methods outperform the rest of the studied machine learning methods. Among the supervised algorithms, XGBoost ranks the first with 97% accuracy, 90.5% Matthews correlation coefficient (MCC), and 99.6% Area Under the Curve (AUC) performance. Moreover, a notable research finding of this study is that the Expectation-Maximization (EM) algorithm, which is an unsupervised method, also performs reasonably well in the detection of the attacks in the NSL-KDD dataset and outperforms the accuracy of the Naïve Bayes classifier by 22.0% .

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**; • **Networks → Mobile ad hoc networks**; • **Computer systems organization → Sensor networks**; • **Security and privacy → Intrusion detection systems**; **Denial-of-service attacks**.

KEYWORDS

Internet of Things, Cybersecurity,Machine Learning,DoS Attacks,Probe Attacks

ACM Reference Format:
Jinxin Liu, Burak Kantarci, and Carlisle Adams. 2020. Machine Learning-Driven Intrusion Detection for Contiki-NG-Based IoT Networks Exposed to NSL-KDD Dataset . In *2nd ACM Workshop on Wireless Security and Machine Learning (WiseML '20), July 13, 2020, Linz (Virtual Event), Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3395352.3402621

WiseML '20, July 13, 2020, Linz (Virtual Event), Austria

ACM ISBN 978-1-4503-8007-2/20/07…$15.00
https://doi.org/10.1145/3395352.3402621

1 INTRODUCTION

Due to rising cybercrime incidents (such as seizing control of industry hardware and smart lights, or recording videos of personal locations), security of Internet of Things (IoT) systems has gained attention in recent years [2]. Several researchers have tackled the applicability of machine learning (ML) algorithms to detect security breaches and attacks on IoT networks. The study in [25] presents a comprehensive survey of ML algorithms for IoT security. Due to the lack of IoT intrusion datasets, a common strategy used in many studies is to use off-the-shelf datasets to inject malicious traffic into IoT networks [18] where the NSL-KDD dataset is directly used to test proposed Intrusion Detection System (IDS). Nonetheless, it is not viable to use a trained model to detect attack patterns in an IoT setting since regular traffic characteristics and attack patterns in IoT networks differ from those in Ethernet-based networks. Researchers explore the available ML algorithms to detect network intrusion datasets in a non-IoT context, and implement routing attacks in IoT environments. However, well-known attacks like DoS and probe still threaten IoT networks.

Contiki-NG is an open source operating system for resource limited devices. It offers the cross-platform benefits, and supports low power communication standards, such as IPv6/ 6LoWPAN, 6TiSCH, RPL, and CoAP. Contiki-NG provides multithreading and optional preemptive multithreading based on protothreads to enhance resource allocation [7]. One of the major contributions of this paper is to examine Contiki-NG when facing attacks from an open benchmark such as NSL-KDD. Thus, UDP, TCP/IP, and 6LOWPAN are carefully investigated to find the exact source that causes the vulnerabilities. The Contiki-NG operating system also provides a simulator called Cooja to help researchers simulate devices and networks, reducing the time and financial cost of experiments. Cooja supports simulation including MAC, network, and application layer protocols, and integration with external tools to provide additional information, such as battery consumption, network interference, and network topologies. This work utilizes Cooja analyzer to collect packet information and PCAP files on a Contiki-NG-based network.

Based on the observations above, we investigate the vulnerabilities in Contiki-NG operating systems in IoT networks, and introduce attack types in the NSL-KDD dataset into a Contiki-NG-based IoT network. In this paper, our methodology is as follows: 1)Introduce possible attacks in the NSL-KDD dataset to the Contiki-NG operating system for IoT nodes; 2) Analyze the vulnerabilities of the Contiki-NG operating system when facing these attacks;3) Generate an IoT network intrusion dataset; 4) Implement an ML-based IDS on the generated intrusion dataset, and analyze the performance of various ML techniques. We study the performance of multiple ML algorithms by training and testing on the dataset in

terms of accuracy, Area Under the Curve (AUC), and Matthews Correlation Coefficient (MCC) metrics. Through numerical analysis, we show that tree-based methods achieve more than 97% accuracy, among which XGBoost ranks the first by outperforming its competitors. Furthermore, among the clustering algorithms, Expectation Maximization (EM) achieves the highest accuracy.

2 BACKGROUND AND RELATED WORK

2.1 Current IoT Intrusion Datasets

Fu et al. [20] report the challenges in obtaining IoT intrusion detection datasets. In addition to the heterogeneous structures in IoT networks, the large scale deployment and distributed topology characteristics of IoT environments also challenge the existing centralized IDS techniques [17]. Due to the lack of public intrusion datasets or benchmarks such as the NSL-KDD dataset, researchers have to set up their unique network topology and generate attacks on it. Some researchers also insert attack records in regular traffic records. Pajaouh, et al. [18] apply the proposed IDS directly on the NSL-KDD dataset. Elike et al. [10] experiment with their IDS on their own IoT network, which contains five nodes, to simulate DDoS attacks. Fu et al. [20] employ Intel-Lab datasets and manually append some records for attacks. Mahmudul et al. [9] leverage Distributed Smart Space Orchestration System (DS2OS) traffic traces; DS2OS is a middleware for storage and brokerage of the state context. In that study, several attacks are also introduced to the environment manually. In [28], Zhang et al. employ Cooja, a Contiki simulator, to generate Distributed Denial of Service (DDoS) attacks and defend by checking consistency. To tackle the heterogeneous attribute of IoT networks, Nadun et al.[19] propose a framework allowing researchers to build their own intrusion dataset by inputting the network packet traffic as raw PACP files. Koroniotis et al. [15] simulate normal traffic using the MQTT protocol in Ubuntu virtual machines, and apply Kali (a operating system designed for penetration testing) to generate BotNet attacks. Argus and Bro-IDS are further used by the authors to extract features. Finally the dataset is fed into three ML algorithms: SVM, Recurrent Neural Network(RNN), and Long Short Term Memory network (LSTM). Gara et al. [8] utilize Cooja as a simulator and simulate the black-hole and gray-hole attacks targeting 6LoWPAN protocols. Instead of using ML algorithms, the authors propose threshold-based solutions to detect such attacks. Sagduyu et al. [21], implement DoS, spectrum poisoning attack, and priority violation attack through adversarial ML algorithms that build on Feed Forward Networks (FFN) to monitor the wireless channel.

Traditional threats like DoS and Probe used in NSL-KDD still threaten the IoT networks, and the current literature: 1) explores the feasibility of ML algorithms under network intrusions datasets that are tailored for non-IoT context, and/or 2) introduces attacks targeting routing protocols in IoT environments. Based on this observations, this paper integrates the conventional attacks in NSL-KDD dataset into an IoT environment, and validates the efficiency of various ML algorithms in this dataset.

2.2 NSL-KDD Dataset

The NSL-KDD dataset has been widely used in numerous studies to validate Network IDS (NIDS) systems and ML algorithms. Four different attack types are present in the NSL-KDD dataset including Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probe attacks. Further attack techniques are used under these four categories. Features in NSL-KDD offer rich information to identify the malicious traffic. For instance, essential features are extracted from the headers of packets revealing the necessary information of packets, content features hold the information of payloads, time-based features offer the analysis of the traffic input over two seconds, and host-based features analyze the behaviour over a series of connections established.

The most commonly used protocols in WSNs are IPv6, TCP, and UDP, whereas protocols such as FTP, mail, SNMP, ARP, and XTerm are not usually seen in WSN environments. Moreover, some of these attacks are designed specially to target Windows and Linux Operating Systems. Thus, these attacks may not be suitable when they are introduced to constrained systems like TinyOS or Contiki-NG. It is particularly worth noting that R2L and U2R are not in the scope of this study since we particularly focus on network layer protocols where user involvement such as login and elevating privileges is not present. More specifically, DoS and probe attacks are the most practical to introduce and test on resource limited devices. Based on these observations, we introduce these types of attack into the simulated environment while multiple ML algorithms are integrated with the IDS to explore their ability to detect these attacks.

3 METHODOLOGY

3.1 IoT Setting Under Study

The architecture of an IoT network used in this paper consists of a group of sensors monitoring a physical environment and aggregating the sensed data for the sink node or gateway. The sink node further delivers the collected data to the cloud servers or to users via a LAN. Since sensors are distributed randomly in the field [22], they are prone to illegitimate access to view and modify the legitimate nodes in the network, or to introduce malicious nodes to the network. Hence, identifying the malicious nodes and classifying the attack types are of paramount importance. Several researchers have studied the possible attacks targeting the RPL protocol, such as blackhole, gray hole, and warm hole, and proposed ML-based detection of these attacks [1]; however, the implementation of 6LoWPAN is still vulnerable to the traditional attacks such as those mentioned in KDD99 or NSL-KDD. With this in mind, we introduce possible network layer attacks in the NSL-KDD dataset to the Contiki-NG-based IoT network, and study the effectiveness of ML algorithms in terms of their classification performance under the NSL-KDD dataset. Below are the attacks considered here:

- **SynFlood**: Malicious nodes keep sending TCP packets with SYN flag creating multiple connections to drain the resources of other nodes [11].
- **Land**: Similar to SynFlood, but source and destination addresses are set as the target node leading the victims to establish TCP connections to themselves [24].
- **UDP Flood**: Malicious nodes randomly send UDP packets to victims, draining their bandwidth and having them send "ICMP port unreachable" messages [12].
- **Ping of Death (PoD)**: PoD is to deliver oversized payload ping requests to victim nodes to cause buffer overflow [26].

- **Smurf**: Malicious nodes broadcast ping requests, setting the source as the victim and leading other nodes to be involved in a flood of ping reply messages to the victims [27].
- **IP sweeping**: Malicious nodes discover active IP addresses by ping requests and enumerating addresses [13].
- **Port sweeping**: Malicious nodes scan the open TCP and/or UDP ports to attack victims [13].

3.2 ML-Integrated Intrusion Detection

Several researchers have studied various ML algorithms on network intrusion detection systems [23]. This study differs from the related work in many ways (as stated earlier), including the type of attacks and the variety of ML algorithms integrated with the IDS system. Eleven ML algorithms are employed, including both supervised and unsupervised ML algorithms. The detection mechanism used in this paper takes the dataset as the input and splits the dataset according to a k-fold (k=10) cross-validation approach to keep the results stable and unbiased. Both training and test datasets are fed into various ML algorithms, and finally multiple evaluation metrics are measured for performance evaluation. The classification methods we test are briefly explained below:

- **Decision Tree (DT)** – A decision tree establishes a tree-like model of decisions in the form of if-else rules. [3]
- **XGBoost** – This is an ensemble learning method with fast training speed and high accuracy. XGBoost is based on DTs, and it leverages gradient boosting and tree-pruning [4].
- **Bagging Tree** – This refers to training multiple weak learners in parallel and aggregating the results in a certain way to avoid overfitting in order to obtain improved results [16].
- **Random Forest (RF)** – An RF combines multiple DTs and selects a subset of training samples and part of the features to prevent overfitting; it finally vote for the best result [16].
- **Bayes Net** – This is a probabilistic graphical model-based technique that builds on Bayesian inference [14].
- **Support Vector Machine (SVM)** – SVM can effectively handle high-dimensional datasets, including the situations where samples are outnumbered by the dimensions [3] .
- **Naïve Bayes** – This is a probabilistic approach that builds on Bayes theorem with the assumption of fully independent features [14].
- **AdaBoost** - This is an ensemble method that can perform oversampling to address the class imbalance problem [16].

In addition to the supervised techniques, clustering (unsupervised) algorithms are also investigated as summarized below:

- **Expectation Maximization (EM)** – The EM algorithm alternates between performing an expectation (E) step and a maximization (M) step to estimate the hidden variables [29].
- **DBSCAN** – This clusters the dataset according to density and is able to find outliers [6].
- **K-Means** – This aims to obtain k clusters out of n data points where each data point is placed in the cluster with the closest mean [6].

4 EXPERIMENTS AND RESULTS

In the simulations, with the settings presented in Table 1, we first implement regular traffic by randomly distributing legitimate nodes.

Table 1: IoT Network Node Types

Node Type	Num	Description
Sink Node	1	Aggregates network traffic and can receives UDP and TCP taffics
Sensor Node (TCP)	10	Simulated as sensor node used for collecting environmental and transfer data via TCP protocol
Sensor Node (UDP)	10	Transfer data via UDP protocol
Malicious Node	1	Simulates the attacker that launches different attacks

Three different kinds of legitimate nodes exist in the simulation: nodes with the UDP protocol; nodes with the TCP protocol; and a sink node which serves as a TCP and UDP server. To simulate realistic WSN traffic, each node randomly sends data with data size that varies according to the TCP and UDP protocols; these nodes are deployed in random locations. Network packets are collected at the sink node by filtering the IP and MAC address. Seven attacks in the NSL-KDD dataset (as also mentioned in Section 3) are implemented in Contiki-NG. Upon the collection of packets, PCAP files are fed into a feature extractor in order to form the intrusion dataset [1].

4.1 Attack Types

We implement seven types of attack techniques from the NSL-KDD dataset, i.e., SynFlood (Neptune), Land, UDP flood, Ping of Death (PoD), and Smurf. Other attacks in the NSL-KDD dataset target the mail protocol, Apache server and Linux systems, such as Mailbomb, Apache2, and Process table. These protocols or tools are not applicable to the Contiki-NG operating system since Contiki-NG is used in resource constrained devices. We further label the dataset according to the MAC address and timestamps. Injection of attacks into the Contiki-NG network is explained below:

- **Syn Flood** – Since Contiki-NG does not provide interfaces to send TCP packets with SYN flag without creating a connection, we directly create IP packets from scratch and send them to the MAC layer protocol. It is worth noting that when the checksum is calculated, the prefix of the IPv6 address is 0xFE80 instead of 0x0000; therefore, using Wireshark for packet analysis will result in several checksum errors. After Syn Flood is introduced into the Contiki-NG network, the sink node keeps sending SynAck to respond to the messages, draining its resource.
- **Land** – As a special Syn Flood attack with identical source and destination IP address in the Contiki-NG system, a Land attack aims to have a node send packets to itself and directly forward these packets to the TCP/IP buffer instead of delivering them to the MAC layer and sending them through wireless channels. Thereby, theoretically, 6LOWPAN could prevent a Land attack by detecting such a situation where the MAC layer delivers a packet with the same source and destination IP address, leaving no chance for such attacks to succeed. This can be observed by monitoring the log of the sink node.

[1] Attack traces are available: http://nextconlab.academy/contikingdata.html

- **UDP Flood** – Unlike a TCP connection, UDP packets do not receive N/ACK from destination nodes. If the destination receives a UDP packet with a closed port, it responds with an unreachable port message (ICMPv6) to the source node. Therefore, in the experiment, UDP packets are sent with random source and destination ports to the sink node, and the sink node responds with ICMPv6 packets immediately. In Contiki-NG, if a source node keeps sending packets with random ports using their UDP interfaces, Contiki-NG refuses to send UDP packets after a few iterations. Hence, in the design of the attack, flooding is performed by generating IP packets and delivering them to the MAC layer. Meanwhile, by doing so, the speed of the flood can be controlled as well.
- **Ping of Death (PoD)** – Contiki-NG checks the size of every message received from a lower layer to prevent buffer overflow. Thus, PoD cannot paralyze or destabilize the sink node but since the sink node will still respond to the ping requests, PoD can be considered as a regular ping flood attack.
- **Smurf** – Smurf is another attack based on ICMPv6, which sends a Ping Request with a broadcast address as the destination and target address as the source IP address. In the simulator, all the nodes send a ping reply to the sink node, blocking other communication.
- **IP Sweeping** – Probes are utilized by adversaries to discover victim IP addresses and ports. In this study, we introduce IP sweeping, TCP port scanning, and UDP port scanning to perform probe attacks. IP sweeping is implemented by iterating on IP addresses while sending ping requests and recording replies.
- **Port Scan** – In TCP port scan, upon completion of the three-way TCP handshake, if the attacker node receives connected events from Contiki-NG, it records the port number. In UDP port scan, if a node sends UDP packets to a closed port, an unreachable port message is replied back by the destination node. Through ICMPv6 events, open ports can be inferred.

4.2 Feature Extraction

The network intrusion dataset creator [19] is utilized to create our dataset. To reduce communication energy, Contiki-NG utilizes the 6LoWPAN protocol at an adaptive layer to compress the length of IP packets as the LibPCAP library does not support the 6LoWPAN protocol. On the other hand, although the Network Intrusion Dataset Creator is also designed for the IPv4 system, it can analyze network packets by using TShark as a backbone, which can also analyze 6LoWPAN protocols. To make it suitable for IPv6, ICMPv6, and 6LoWPAN protocols, we modify the feature extraction module and add more features related to the ICMPv6 protocol, IP address, and MAC address. Finally, twenty-eight features are extracted from the network flow, and each entry also has a label denoting the the type of the attack introduced through the corresponding packets. The extracted features can be categorized as follows:

- **Frame Info** – The total physical frame length of TCP, UDP, ARP, and ICMPv6 packets, and the number of packets.

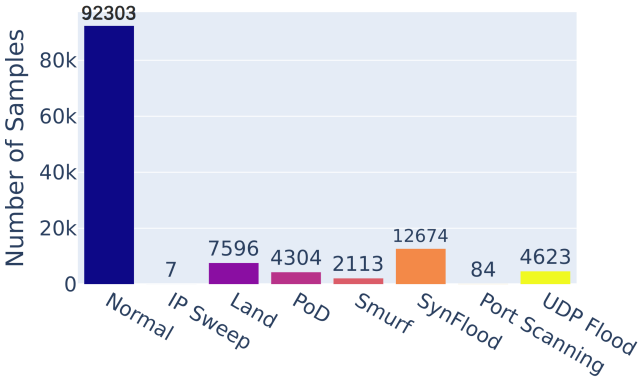

Figure 1: Dataset Distribution

Table 2: Accuracy, Precision, Recall and F-Scores

	Accuracy	Precision	Recall	F Score
RF	0.966	0.969	0.967	0.968
DT	0.966	0.969	0.967	0.968
Bagging	0.967	0.969	0.967	0.968
SVM	0.957	0.948	0.957	0.951
NB	0.452	0.904	0.452	0.545
BN	0.882	0.944	0.882	0.902
AdaBoost	0.740	0.663	0.740	0.646
XGBoost	**0.970**	**0.970**	**0.968**	**0.968**

- **IP Layer Info** – The total IP frame length of TCP, UDP, ARP, and ICMPv6 packets, the number of connection pairs, the number of ICMPv6 types, and the number of ARP and ICMPv6 packets.
- **IP Address Info** – Whether the source and the destination have the same IP address, and whether the IP address is consistent with MAC address.
- **Transport Layer Info** – The total frame length of TCP, UDP, and the number of ports used in both TCP and UDP packets.
- **Application Layer Info** – The number of SSL, HTTP, FTP, SSH, SMTP, DHCP, and DNS packets.

To better understand the generated dataset, we present a breakdown of the packets in the Contiki-NG network in Fig 1. Since most of the time (24 hours) the nodes in the simulator send regular traffic to the sink node, the normal packets form the majority class. Due to the limitation of the hardware and simulator, each DoS attack only runs for 5 minutes. In the simulation, we also implement IP sweep and port scan, but the available IP address is obtained alongside the open port number, so such attacks will be blocked. Hence the sample size of probe attacks is negligible, especially for IP sweep. Since the generated dataset is significantly imbalanced, techniques and comparison metrics that can handle imbalanced datasets are used to train and evaluate the results.

Table 3: TP, RF, MCC, and AUC of ML Algorithms

	TP Rate	FP Rate	MCC	AUC
RF	0.966	0.046	0.903	0.995
DT	0.966	0.048	0.903	0.993
Bagging	0.967	0.048	0.904	0.995
SVM	0.957	0.072	0.865	0.942
NB	0.452	0.028	0.407	0.888
BN	0.882	**0.006**	0.789	0.985
AdaBoost	0.740	0.065	0.038	0.604
XGBoost	**0.970**	0.046	**0.905**	**0.996**

4.3 ML Algorithms to Classify Intrusions

We employ the following ML algorithms to classify and cluster intrusions: AdaBoost, Random Forest, Decision Tree, Bagging, XGBoost, SVM with RBF kernel, Naïve Bayes, Bayes Network, KMeans, and DBSCAN. For supervised methods, accuracy, true positive rate (TP), false positive rate (FP), precision, recall, F measure, Matthews correlation coefficient (MCC), and Receiver Operating Characteristics area under the curve (AUC) are used. MCC is calculated as $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}}$, and is a useful ML algorithm metric especially for imbalanced datasets, which results in more meaningful results than F score [5] in such settings as it considers the proportion of the classes inside the confusion matrix.

As for the unsupervised methods, we simply use accuracy to present the result, since the results of clustering algorithms still need to be enhanced. As mentioned earlier, the generated dataset is imbalanced; hence accuracy itself is not enough to represent the performance, while F measure, MCC, and ROC area can better show the results of these ML algorithms. All the results of supervised methods discussed in this paper use ten fold cross-validation to generate the final performance values.

Performance results of classification algorithms are listed in Table 2 and Table 3. Initially, Adaboost and RUSBoost were selected as strong ensemble methods to operate with imbalanced data. However, as shown in Table 2 and 3 the performance of these algorithms is not strong enough; in particular, the RUSBoost underperforms compared to AdaBoost. Therefore, only AdaBoost performance is shown in the performance results. Other ensemble methods are also tested such as Bagging, Random Forest (RF), and Extreme Gradient Boosting (XGBoost), among which XGBoost outperforms the others in terms of all performance metrics. RF and Bagging techniques almost have identical performance, which is slightly higher than that of a single decision tree. Considering the limitation of nodes in IoT or WSNs, (i.e., limited RAM, storage capability, and energy resource), decision trees stand out as the best fit for an IoT environment. Furthermore, DT is highly explainable so manual tuning of the trained model is also possible and suitable for specific use cases in real-life. Last but not least, the performance of decision trees is reasonably close to that of a more advanced method, XGBoost.

Different kernels of SVMs are tested and Radial Basis Function (RBF) outperforms the other functions. However, the performance of SVM with RBF is still outperformed by the tree-based methods. Furthermore, SVM requires an extensive amount of time to train and run. Naïve Bayes (NB) and Bayes Network (BN), due to their probabilistic nature, are not suitable for this problem, and perform poorly as seen in the table.

Table 4: Performance of Unsupervised Algorithms

	Accuracy	Clusters	Identified Classes
K-Means	0.496	6	Normal, Land, PoD, SynFlood, UdpFlood, Smurf
DBSCAN	0.511	4	Normal, Land, SynFlood
EM	0.672	4	Normal, Land, PoD, UdpFlood

This paper also utilizes unsupervised methods to validate their performance and reports them in Table 4. Regardless of tuning of the parameters, the performance of unsupervised methods is limited. For K-Means, when the number of clusters is set to six, the overall accuracy is 49.59%. Furthermore, when the parameters of DBSCAN are set to eps=0.4 and minPts=6, its accuracy is 51.1%. Expectation-Maximization (EM) stands out among the unsupervised techniques, offering the best results among unsupervised algorithms with 67.2% accuracy, which outperforms Naïve Bayes and approaches the performance of AdaBoost. Although the unsupervised methods underperform when compared to supervised ML techniques such as the tree-based methods, they do not require labels (which generally cannot be easily generated in real-life networks). In such a scenario, the EM algorithm can be considered as a potential solution..

5 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we introduce possible DoS and probe attacks in the NSL-KDD dataset to an IoT network, specifically Routing Protocol for Low-Power and Lossy Networks (RPL) and 6LoWPAN networks, using the Contiki-NG operaing System. Moreover, the generated dataset is fed into eleven ML algorithms to explore their ability to classify different attacks. Tree-based methods and ensemble algorithms such as XGBoost, Decision Trees (DTs), Bagging Trees, and Random Forest perform well and achieve more than 96% accuracy, whereas the remaining methods, Bayes Network, Naive Bayes (NB), Adaboost, perform relatively poorly. Considering that the IoT devices consist of resource constrained devices, DTs, as a simple and highly explainable solution with high performance, is concluded to be utilized in the sink node to detect DoS and probe intrusions. Moreover, when unsupervised methods are utilized (EM, DBSCAN, and K-Means), DBCAN and K-Means are outperformed by EM while the EM algorithm outperforms NB by 22% in terms of accuracy. Therefore, EM is concluded to be more suitable for the NSL-KDD dataset and the Contiki-NG-based IoT scenario when it is not possible to acquire the labels from the network traffic. Therefore, tests under unsupervised methods are planned to be extended to implement and test Hidden Markov Model, Gaussian Mixture Model, and other EM-based algorithms.

ACKNOWLEDGMENTS

This study has been supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under the DISCOVERY program.

REFERENCES

[1] Iman Almomani, Bassam Al-Kasasbeh, and Mousa AL-Akhras. 2016. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks.

[2] Hazim Almuhimedi et al. 2015. Your Location has been Shared 5,398 Times! A Field Study on Mobile App Privacy Nudging. In *ACM Conf. Human Fact. in Comp. Sys.* Seoul, Korea, 787–796.

[3] Ethem Alpaydin. 2020. *Introduction to Machine Learning* (fourth edition edition ed.). The MIT Press, Cambridge, MA.

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *ACM SIGKDD Int Conf. on Knowledge Discovery and Data Mining*. San Francisco, CA, USA, 785–794.

[5] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21 (Jan. 2020).

[6] G. Dong, Y. Jin, S. Wang, W. Li, Z. Tao, and S. Guo. 2019. DB-Kmeans:An Intrusion Detection Algorithm Based on DBSCAN and K-means. In *20th APNOMS Symposium*. 1–4.

[7] A. Dunkels, B. Gronvall, and T. Voigt. 2004. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks*. 455–462. ISSN: 0742-1303.

[8] Fatma Gara, Leila Ben Saad, and Rahma Ben Ayed. 2017. An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs. In *Int. Wireless Communications and Mobile Computing Conf.* 276–281.

[9] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, and M. M. A. Hashem. 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things* 7 (Sept. 2019), 100059.

[10] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis, and Robert Atkinson. 2016. Threat analysis of IoT networks using artificial neural network intrusion detection system. In *Int. Symp. on Networks, Comp. and Comm.* Yasmine Hammamet, TUN, 1–6.

[11] Khalid Hussain, Syed Jawad Hussain, NZ Jhanjhi, and Mamoona Humayun. 2019. SYN Flood Attack Detection based on Bayes Estimator (SFADBE) For MANET. In *Int. Conf. on Computer and Information Sci. (ICCIS)*. 1–4.

[12] Kamaldeep, M. Malik, and M. Dutta. 2017. Contiki-based mitigation of UDP flooding attacks in the Internet of things. In *Int. Conf. on Comp., Comm. and Automation*. 1296–1300.

[13] Nattawat Khamphakdee, Nunnapus Benjamas, and Saiyan Saiyod. 2014. Improving Intrusion Detection System based on Snort rules for network probe attack detection. In *Int. Conf. on Information and Communication Technology*. 69–74.

[14] K. C. Khor, C. Y. Ting, and S. Phon-Amnuaisuk. 2010. Comparing Single and Multiple Bayesian Classifiers Approaches for Network Intrusion Detection. In *Int. Conf. on Computer Engineering and Applications*, Vol. 2. 325–329.

[15] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. 2019. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gen. Comp. Sys.* 100 (Nov. 2019), 779–796.

[16] N. Lower and F. Zhan. 2020. A Study of Ensemble Methods for Cyber Security. In *Computing and Communication Workshop and Conf.* 1001–1009.

[17] Marc-Oliver Pahl and François-Xavier Aubet. 2018. All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection. In *14th International Conf. on Network and Service Management (CNSM)*. 72–80. ISSN: 2165-9605.

[18] Hamed Haddad Pajouh, Reza Javidan, Raouf Khayami, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2019. A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Trans. on Emerging Topics in Computing* 7 (April 2019), 314–323.

[19] Nadun Rajasinghe, Jagath Samarabandu, and Xianbin Wang. 2018. INSECS-DCS: A Highly Customizable Network Intrusion Dataset Creation Framework. In *IEEE CCECE,*. 1–4.

[20] Rongrong Fu, Kangfeng Zheng, Dongmei Zhang, and Yixian Yang. 2011. An intrusion detection scheme based on anomaly mining in Internet of Things. In *IET Int. Conf. on Wireless, Mobile & Multimedia Networks*. 315–320.

[21] Yalin E. Sagduyu, Yi Shi, and Tugba Erpek. 2019. IoT Network Security from the Perspective of Adversarial Deep Learning. In *IEEE Int. Conf. on Sensing, Commu., and Networking*. 1–9.

[22] Mustapha Réda Senouci, Abdelhamid Mellouk, and Amar Aissani. 2014. Random deployment of wireless sensor networks: a survey and approach. *Int. J. Ad Hoc Ubiquitous Comp.* (2014).

[23] Muhammad Shafiq, Zhihong Tian, Yanbin Sun, Xiaojiang Du, and Mohsen Guizani. 2020. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems* 107 (June 2020), 433–442.

[24] Subrina Sultana, Sumaiya Nasrin, Farhana Kabir Lipi, Md Afzal Hossain, Zinia Sultana, and Fatima Jannat. 2019. Detecting and Preventing IP Spoofing and Local Area Network Denial (LAND) Attack for Cloud Computing with the Modification of Hop Count Filtering (HCF) Mechanism. In *Int. Conf. on Comp., Comm., Chemical, Materials and Electronic Eng. (IC4ME2)*. 1–6.

[25] Syeda Manjia Tahsien, Hadis Karimipour, and Petros Spachos. 2020. Machine learning based solutions for security of Internet of Things (IoT): A survey. *Journal of Network and Computer Applications* 161 (2020), 102630.

[26] Fekadu Yihunie, Eman Abdelfattah, and Ammar Odeh. 2018. Analysis of ping of death DoS and DDoS attacks. In *IEEE Long Island Sys., Applications and Technology Conf.* 1–4.

[27] Gholam Reza Zargar and Peyman Kabiri. 2009. Identification of effective network features to detect Smurf attacks. In *IEEE Student Conf. on Research and Development*. 49–52.

[28] Congyingzi Zhang and Robert Green. 2015. Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network. In *18th Symposium on Communications & Networking*. Alexandria, VA, 8–15.

[29] Yüksel Öner and Hasan Bulut. 2020. A robust EM clustering approach: ROBEM. *Communications in Statistics - Theory and Methods* 0, 0 (Feb. 2020). Publisher: Taylor & Francis.