ISSN 2278-3091
Volume 10, No.1, January - February 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse161012021.pdf
https://doi.org/10.30534/ijatcse/2021/161012021

EADA: An Algorithm for Early Detection of Attacks on IoT Resources

Jalindar Karande[1], Prof. Sarang Joshi[2]
[1,2]Department of Computer Engineering,
Pune Institute of Computer Technology,
Savitribai Phule Pune University, Pune, India
Jalindar.karande@ieee.org

ABSTRACT

IoT applications are becoming widespread in monitoring and managing critical infrastructure. Many attacks have been demonstrated in the state-of-the-art on IoT resources. These attacks make use of vulnerabilities present in various connected systems and the Internet of Things (IoT). The state-of-the-art presents many approaches to detect and mitigate such attacks on IoT resources. The early attack detection mechanism is essential to prevent damage to the IoT system and human. This paper presents an algorithm for early detection of attacks on IoT resources through use of predictive descriptor tables. Effectiveness of the proposed algorithm is evaluated through experimental setup built using Google cloud platform. Experimental results show that the proposed algorithm is efficient in the detection of attacks in real-time.

Key words: IoT, IoT Security, Cloud Computing, Early Detection.

1. INTRODUCTION

The Internet of Things (IoT) is being accepted in every business domain including but not limited to energy, healthcare, construction, oil and gas, manufacturing, transportation and agriculture. These widespread applications are motivated by the capability of IoT systems to interconnect and communicate without human intervention. This capability of the IoT system makes it an attractive solution for continuous monitoring, taking autonomous decisions and controlling connected systems. These solutions reduce dependency on human intervention and human errors. Such IoT systems provided real-time surveillance, support and helped to keep critical infrastructure functioning during recent COVID-19 outbreak when most of the human were locked down. Applications of IoT devices during outbreak includes connected thermometers for identifying COVID-19 hotspots, connected wearable for monitoring health status and connected robots for disinfecting the high-risk area.

An Autonomous nature of IoT systems communications and widespread unattended use makes it open to various ways of exploits. Any malicious modification to data collected by IoT devices may result in critical failures of the connected system impacting business as well as human lives. In some applications, if data is captured by an unauthorized person may lead to privacy concerns. Attacks on IoT devices results in unavailability of timely data from critical cyber-physical systems. Such unavailability data for critical decision making may result in a catastrophic situation. This highlights the need for security of data communication within IoT systems as well as security of IoT resources.

The need of real-time analytics and basic experimental demonstration of IoT security is presented in [1]. This paper presents a detailed algorithm for identifying inter-system attacks in IoT systems. This paper also presents a comparative analysis of the performance of the proposed system in terms of detection accuracy. This paper also includes an analysis of datasets and presents feature importance along with the impact of feature engineering on it.

The next section presents a review of the state-of-the-art of IoT security needs and implementation mechanisms. Section III presents the proposed mechanism for real-time detection of attacks on IoT resources along detailed algorithm Section IV demonstrates the experimental setup used for the implementation of real-time security attack detection using a Google cloud platform. Section V presents an analysis of the results of the proposed algorithm and its comparison with the state-of-the-art.

2. LITERATURE REVIEW

Challenges in implementing security measures like authentication, authorization and access control in IoT devices include low computing power, heterogeneity of devices and vulnerabilities in deployment platforms [2]. These vulnerabilities in devices and platforms are exploited to launch an attack against IoT resources. Irrespective of the presence of attack prevention mechanisms in IoT devices, the resource-constrained nature of device and casual approach by device owner can still make it vulnerable to attacks. The vulnerability assessment of consumer IoT devices is presented in [3] highlights that 10% of consumer devices had critical risk vulnerabilities. It further highlights that 40% of the assessed devices had high-risk vulnerabilities present. The devices included in the assessment include a webcam, smart TV and printers from a wide range of manufacturers. This highlights the importance of detection of the presence of

attack/vulnerability even though preventive measures are taken.

Security attack detection mechanisms and their effectiveness are evaluated in various scenarios like a single attacker, multiple attackers and collaborative attackers in [4]. The attacks on the IoT devices are broadly classified according to target vulnerabilities in a) Device-based b) Network-based and c) Software-based. The detailed study on ways attackers can exploit IoT devices, mitigation approaches is presented in [5].

Behaviour-based analysis of the vulnerability of drone-based IoT system along with detection of vulnerability using Petri net is presented in [6]. Attackers exploit vulnerabilities in IoT devices and protocols to enter into the IoT networks. The approach based on modelling relationship between vulnerabilities as a graph and using the graph-theoretic approach for detecting attack is presented in [7].

A game theory-based approach is for attack detection along with reputation model is presented in [8], which is capable of detecting various attacks on IoT systems. A review Game theory-based approach for detection of attacker considering conflicting goals of attackers and detection engines is presented in [9] to detect security attacks in IoT systems.

Intrusion detection in IoT trough traffic filtering is presented in [10]. This work also highlights several open challenges in attack detection using traffic filtering which includes complex traffic characterization, difficulties in preparing blacklist and white list for traffic filtration, traffic sampling, building realistic attack models and impact of false positives. Deep packet inspection based attack detection mechanism is presented in [11]. This mechanism makes use of regular expression in terms of DFA to represent the rule. Representation of rules in regular expression makes it easy to implement in the hardware through Field Programmable Gate Arrays (FPGAs) which make it faster than software approaches. The number of states in regular expression required to represent all possible attack signatures is very large, and there are always chances of changing signature by new attacks.

A review of machine learning-based approaches for enhancing the security of IoT system is presented in [12]. These approaches include authentication based on the prediction of communication parameters, machine learning algorithms for access control, secure offloading and machine learning-based attack detection methods. This paper further concluded that machine learning needs intensive computing power and high communication overhead. Also, the need for a large amount of training data and complex feature extraction process makes these algorithms non-attractive for resource-constrained devices.

Random Neural Network-based approach for detection of attackers in IoT systems is presented in [13]. This approach learns anomalies in the performance of the system using the Random neural network and relates it to the failure of IoT node or attacker's presence. Deep learning-based approach for attack detection in IoT is presented in [14]. A framework for DDoS attack detection in IoT systems based on cosine similarities within the traffic is presented in [15]. Artificial neural network-based architecture for detection of DDoS/DoS attack is presented in [16]. This architecture makes use of both forward and backward learning mechanisms to train and identify malicious traffic.

The BAT model with multiple convolutional layers is presented in [17] combines attention and bidirectional long short-term memory to obtain key features from packet stream to identify an attack. Another deep learning-based model includes recurrent neural network-based model called RNN-IDS presented in [18] and Convolution Neural Networks (CNN) based model is presented in [19]. Offline machine learning-based methods usually perform well because they have access to the entire dataset during the training phase and can perform multiple iterations over the data for better training. However, they are not effective when computational power and memory are limited.

Hidden Markov model-based classifier is proposed in [20] to detect anomalies in data, which is used to alert about security attack. This method makes use of multiple knowledge domains like physical process knowledge and control system knowledge to identify an attack. This approach is suitable for implementation in an industrial control system, but may not be suitable for resource-constrained IoT systems. Machine learning-based mechanism using inference and predicting states of the system is presented in [21] to detect anomalies and attacks in the IoT systems.

The model for distributed detection of security attack on IoT system is presented in [22] along with proof of the concept implementation, which make use of fog computing nodes to deploy extreme learning machine based mechanism for attack detection at local. Further security state information collected from fog computing node is summarized at a cloud node to identify the next action of the attacker.

The key difference between the implementation of security in IoT systems and other conventional systems is the heterogeneity in IoT resources and network protocols along with a big volume of data being generated. Public cloud platforms are being extensively used in IoT systems to deal with big data [23], [24]. The need for building real-time attack detection mechanism using public cloud-hosted big data platforms and using data stream-based machine learning algorithms for timely and accurately detect attack is the need of the time[25], [26].

Mechanisms for detection of security attacks on IoT resources presented in the state-of-the-art are evaluated with batch processing on the dataset. For real-time and early attack detection, we need to process a live data stream from IoT devices. Attack detection mechanism on such real-time data stream is missing in the state-of-the-art to the best of author's knowledge. The need for End-to-End live attack detection mechanism for IoT devices is highlighted in the state-of-the-art, but the demonstration of such End-to-End deployment and evaluation of its effectiveness is missing in the state-of-the-art to the best of author's knowledge. This

motivated authors to design a real-time mechanism for detection of attacks on IoT resources and analyze its efficiency.

3. DATASET ANALYSIS

KDD-99 is one of the first benchmark datasets in network security research domain. This dataset consists of network traffic from attack and normal scenario. Several pieces of research in state of the art question its effectiveness for building network attack detection models because of the lack of the presence of recent attacks [27]. It also has a more skewed distribution of the traffic. NSL-KDD is a subset of KDD-99 data set designed to overcome challenges presents in the KDD-99 data set. This dataset removes a large portion of redundant records present in KDD-99 to avoid skewed distribution. This dataset still lacks the presence of traffic traces from attacks recently demonstrated [28].

The ISCX-IDS [29] dataset consists of real network traffic captured which includes traffic class labels and the complete network packet payload. CICIDS2017 [30] dataset consist of synthetically created network traffic which includes records from normal and recent attacks scenario. ISCXTor2016 [31] designed by collecting real network traffic from real users, while users were using browser-based applications, FTP, P2P applications and email systems.

The BoT-IoT dataset[32] is built by collecting traffic traces from testbed deployed in the laboratory, which consists of traffic from normal nodes and botnets. A UNSW-NB15 [33] is built by collecting traffic traces from experimental setup in the laboratory consist of attacks and legitimate nodes. N-BaIoT[34] is another dataset build by collecting network traffic traces from laboratory testbed. This testbed includes traffic from IoT devices demonstrating. This dataset consists of attacks specifically on IoT devices.

NSL-KDD and N-BaIoT are selected for experimental setup further as the most cited dataset and most recent dataset respectively. Further NSL-KDD is transformed into JSON and AVRO format from exiting text format because JSON and AVRO are recommended formats for big data processing. IoT devices send data in JSON format, so using dataset in JSON format makes experimental setup closer to real life. Dataset generated as part of this experimental setup are available at [35] and [36].

Figure 1 Shows feature importance for top 20 important features in terms of weight coefficients of attack detection generated using NBaIoT dataset without feature engineering. Figure 2 and Figure 3 shows the frequency distribution of HH_L3_pcc and HH_jit_L0_01 mean respectively, during normal and anomaly situation. From Figure 3 and Figure 4, it is evident that HH_jit_L0_01 is more useful for differentiating attacker from normal nodes. Feature importance presented in Figure 1 simply indicates that HH_jit_L0_01 is not significant. Similarly, HH_L3_pcc is shown in Figure 1 as a highly important feature, but its frequency distribution presented in Figure 2 shows it is not significant for classification. These figures highlight that weight coefficient and real importance of feature are not correlated unless features are normalized.

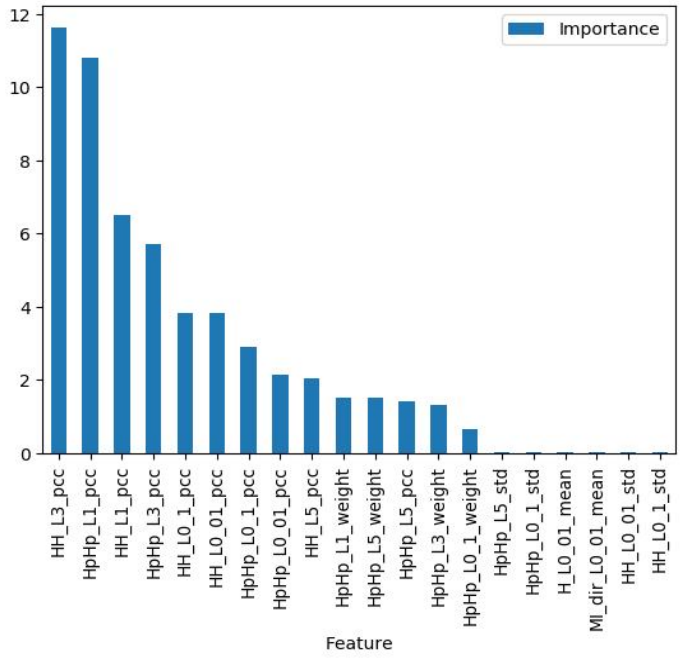

Figure 1: Feature Importance on N-BaIoT dataset without feature normalization

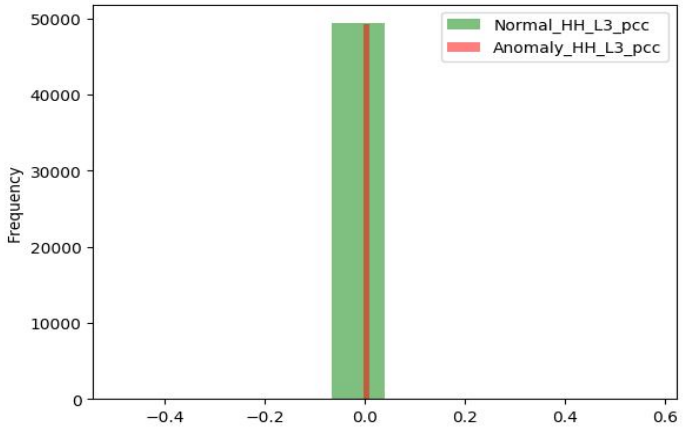

Figure 2: Frequency distribution off HH_L3_PCC in normal and attack scenario

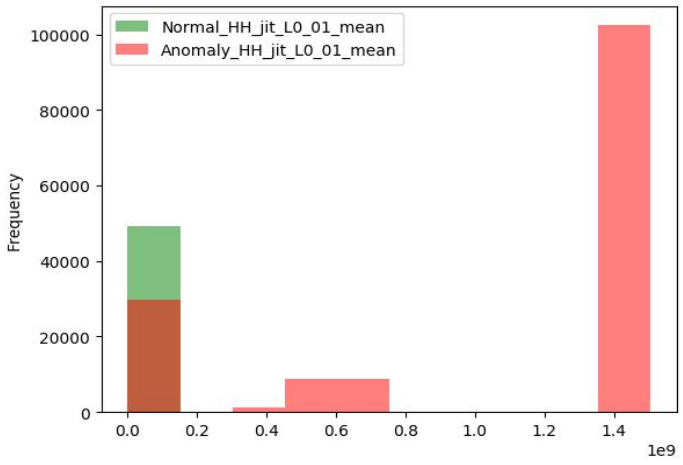

Figure 3: Frequency distribution off HH_jit_L0_01_mean in normal and attack scenario

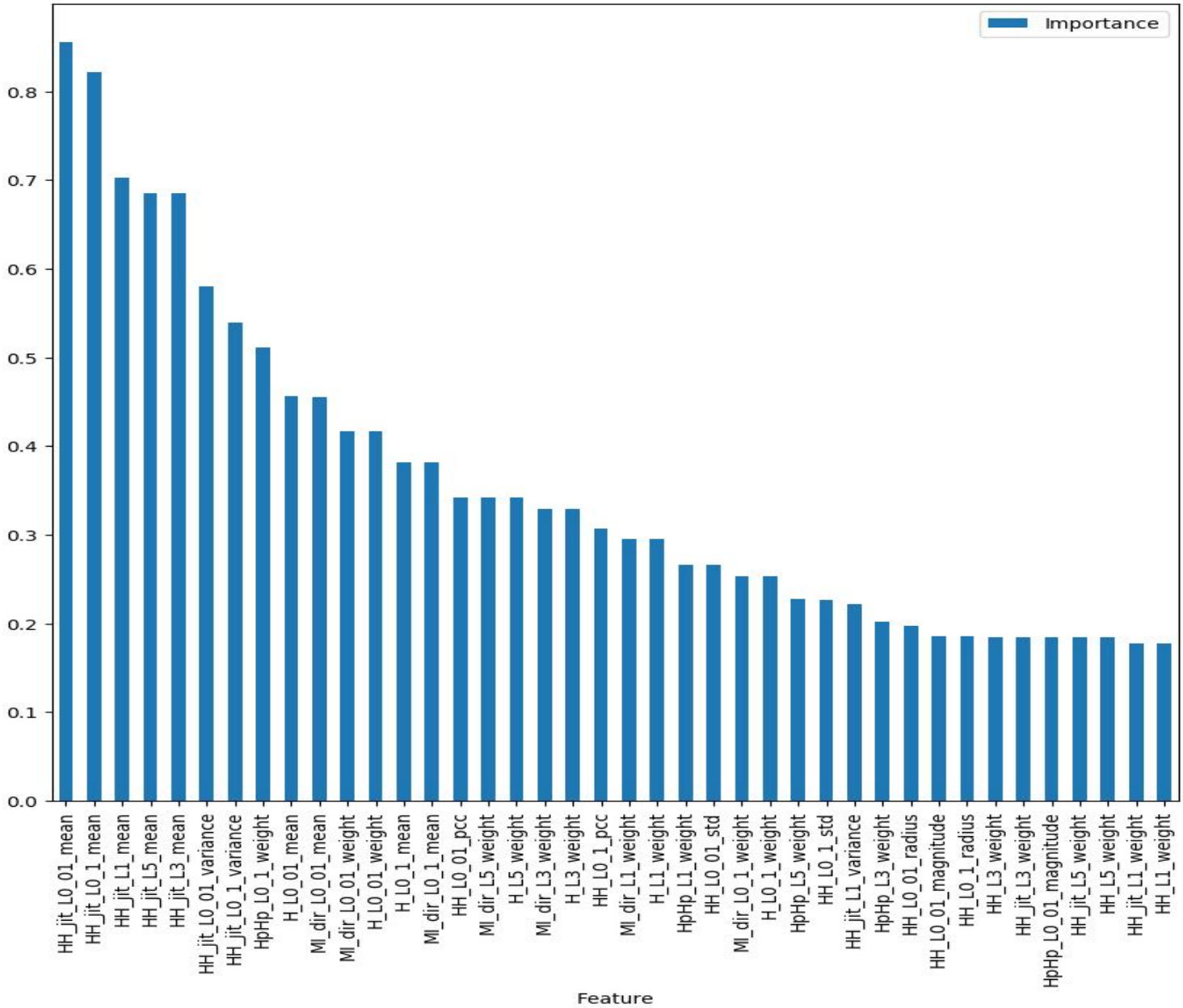

Figure 4: Feature Importance on N-BaIoT dataset after feature normalization

Figure 4 shows feature weight coefficients after normalization of all features. This shows clear importance of features and also in-line with frequency distribution. Figure 5 shows the feature importance from NSL-KDD dataset. This motivates us to normalize features generated in real-time before using them for attack detection. The detailed algorithm for real-time feature normalization is presented in the next section.

4. PROPOSED SOLUTION

Figure 6 shows the block diagram of the proposed architecture. In the proposed architecture, Gateway node is the entry point for the IoT network in case of multi-hop IoT system. In the case of single-hop IoT devices, every device will act as a gateway node. Gateway node holds additional responsibility of forwarding headers of all packets received to stream processor using MQTT protocol [37]. The stream processor works as a publish-subscribe broker and responsible for queue management and reliability. The stream processor is also responsible for maintaining the list of topic corresponding to fields in the packet. The stream processor is responsible for handling backpressure if any. Real-time feature generators subscribe to topics on stream processor.

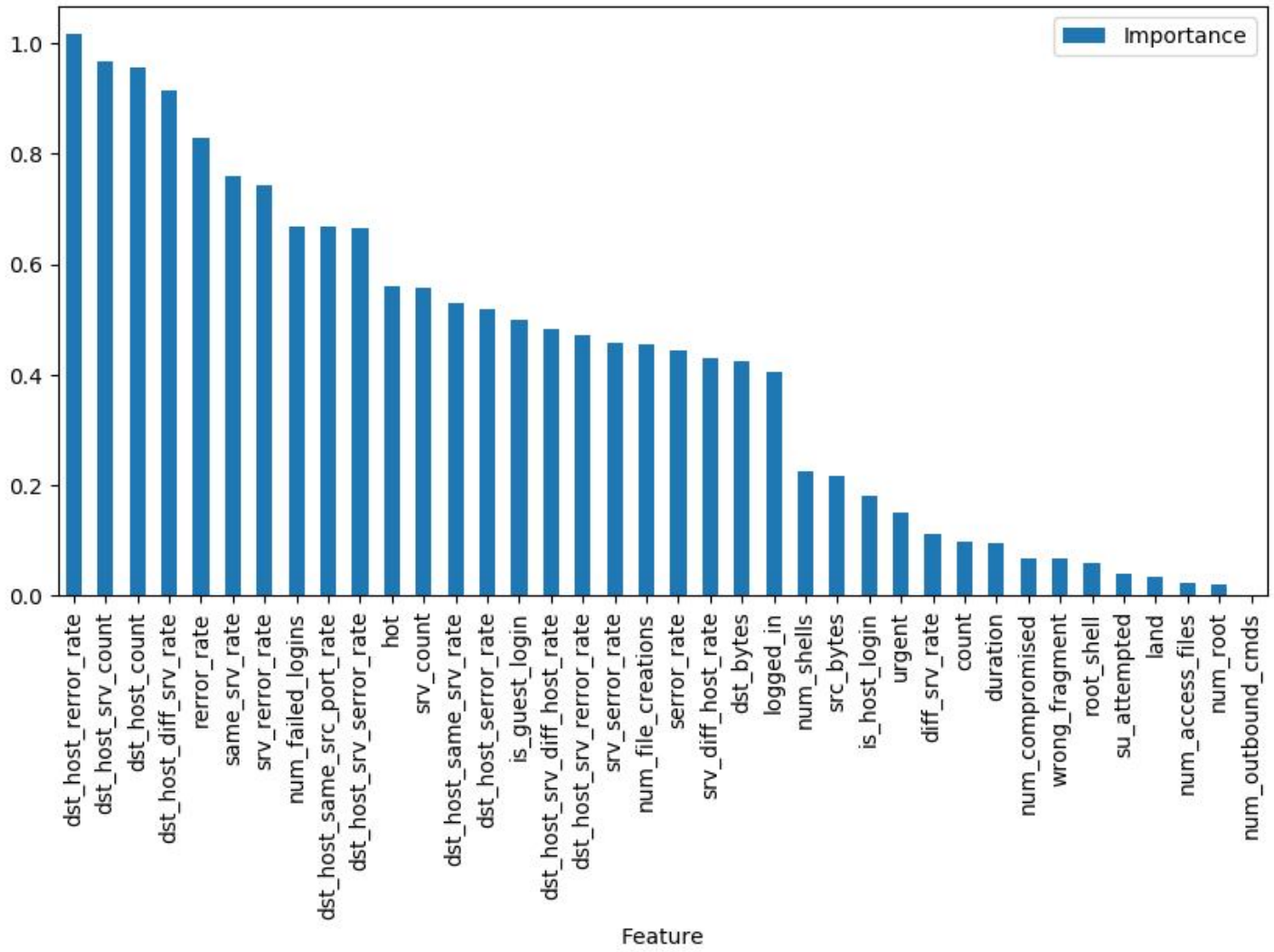

Figure 5: Feature Importance on NSL-KDD dataset after feature normalization

Every feature generator takes some predefined part of the packet header based on the topic of subscription and generates features based on it in real-time. These real-time features are used to building a predictive descriptor table. Predictive descriptor tables hold current and past values for every feature. Values for the feature in future are predicted based on the trend of changes in values. Predictive descriptor tables used to calculate score called PSAS (Packet source abnormality score), which indicate the probability of the source being attacker. This PSAS score is used as a measure for predicting the behaviour of the source of the packet. The detailed algorithm for early attack detection is presented in Algorithm 2.

The formula for calculation of PSAS score is defined in "(1)"

$$SAS = \sum_{j=0}^{m} c_j * (f_{jt0} + \sum_{i=0}^{n} (f_{ji} - f_{jt(i-1)}$$

(1)

Where, m indicates the number of features in descriptor table, C_j indicate the coefficient of j^{th} feature, f_{jti} indicate a value of j^{th} feature at the time i and n indicate the number of records present in the descriptor table. Here coefficient reflects feature importance values presented in Figure 3 and Figure 4.

PSAS is compared with the threshold value to detect the attacker. The default value of the threshold is 0.5. Value of threshold will be decided based on a tradeoff between false positive rate and false-negative rate. The detailed algorithm for run time feature generation is shown in Algorithm 1. Every Real-time feature generator subscribes to a certain set of topics from the MQTT broker. Every categorical column is label encoded to convert it to numeric values, and it is further one hot encoded to use in calculating PSAS. All numerical features, including one, converted from categorical to numerical go through normalization to convert them in the standard range. This avoids PSAS value being wrongly influenced by certain feature with broader domain range. Mean and standard deviation used for normalization calculations are updated at the run time.

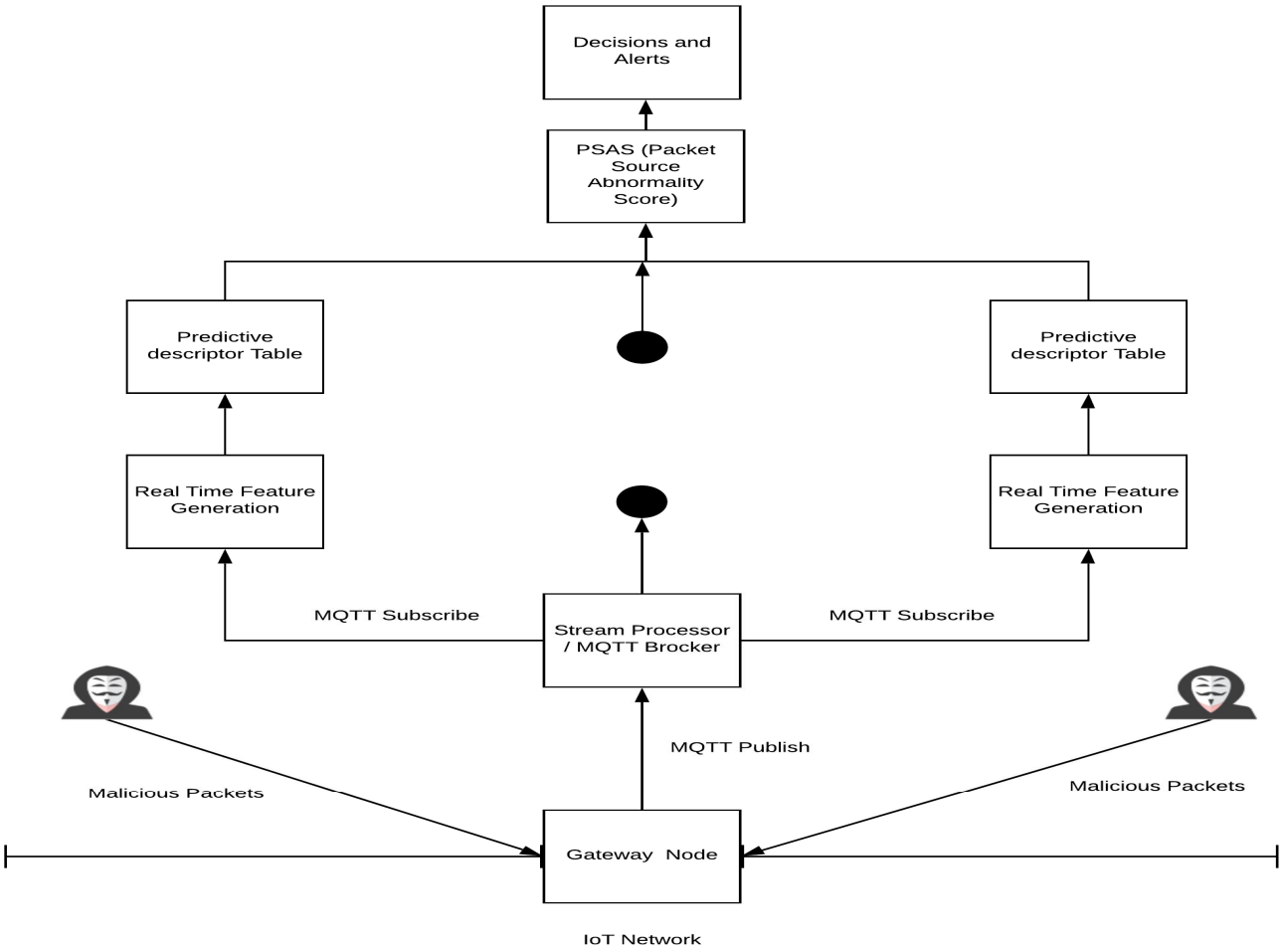

Figure 6: Block diagram of the proposed solution

Algorithm 1: Real-time feature Generation

Result: Real-time features values V
Let C is set of columns ;
for each column *c* in C **do**
  Let μ mean of c;
  Let σ standard deviation of c;
  Let v value of column c in V;
  if v *is categorical* **then**
    v ←LabelEncoder(v);
    v ←OneHotEncoder(v);
  end
  c=(v- μ)/ σ;
  Update μ;
  Update σ;
End

Algorithm 2: Early Attack Detection Algorithm

Result: Attack detection
Let F is set of Features;
PSAS ←0;
for each Feature f in F **do**
  Let H is set of historical values;
  V←current value of feature f
  for each h in H **do**
    if v(h-1) *is exist* **then**
      V← V+(v(h) –v(h-1)
    else
      V← V+(v(h))
    end
  end
  PSAS ←PSAS +w(f)*V
end
if PSAS > = threshold **then**
  return attack
else
  return normal
end

5. RESULT ANALYSIS

The experimental setup for performance analysis of the proposed method is implemented using tools available on the public cloud platform on Google. Google Cloud Platform (GCP) tools used the setup includes Cloud Pub/Sub, Cloud Dataflow and BigQuery. Cloud Pub/Sub is the publish-subscribe broker and has high scalability, availability and throughput. Cloud Pub/Sub is responsible for managing topics and their subscribers. Cloud Dataflow in the experimental setup will act as a real-time feature generator and connector between Cloud pub/sub and BigQuery. Cloud Dataflow is a fully managed and serverless parallel processing engine. This parallel processing helps generate features in parallel and to avoid latency. Combination of Cloud Pub/Sub and Cloud Dataflow helps handle big volume and velocity of data from a large number of devices in parallel with low latency and no data loss.

BigQuery is another fully managed serverless service from GCP platform. In the experimental setup, BigQuery is responsible for building predictive descriptor table, calculating PSAS score and detecting attacker in real-time. BigQuery also has its machine learning libraries called BigQuery Machine Learning (BQML). This makes use of SQL for training and evaluating machine learning models though numbers of algorithms are limited.

Accuracy of attack detection of the proposed method based on traffic generated using NSL-KDD dataset is evaluated. Accuracy of the proposed algorithm is compared with other algorithms presented in the state-of-the-art on the same dataset and presented in Figure 7. The comparative analysis shows that the proposed method outperforms the other algorithm presented in the state-of-the-art.

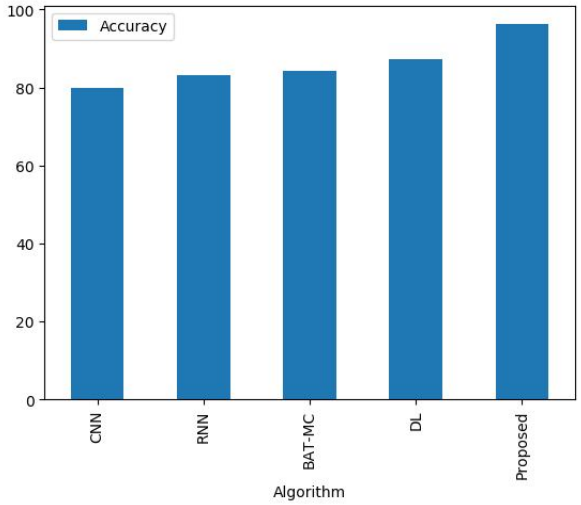

Figure 7: Comparison of accuracy of the proposed algorithm with CNN [19], RNN [18] BAT-MC [17] and DL [18] on NSL-KDD dataset

6. CONCLUSION

This paper highlighted the need for early attack detection on IoT resources. This paper presented the mechanism for early attack detection and its implementation on the public cloud platform. The result analysis of the proposed method shows improvement in performance through the use of parallel feature generation and application of predictive descriptor tables. Our future work includes designing an algorithm for detecting collaborative attacks in early stage.

REFERENCES

[1] J. Karande et al., "Real-Time Detection of Cyber Attacks on the IoT Devices," Computing, Communication and Networking Technologies (ICCCNT), 2020 11th International Conference on, IEEE, pp. 1-6, 2020, doi: 10.1109/ICCCNT49239.2020.9225487.

[2] Z.-K. Zhang et al., "IoT security: ongoing challenges and research opportunities," in 2014 IEEE 7th international conference on service-oriented computing and applications. IEEE, pp. 230–234,2014, doi:10.1109/SOCA.2014.58.

[3] R. Williams et al., "Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach," in 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 179–181, 2017, doi:10.1109/ISI.2017.8004904.

[4] J. Karande et al., "Comprehensive Assessment of Security Attack Detection Algorithms in Internet of Things," in Computing Communication Control and Automation (ICCUBEA), 2018 Fourth International Conference on. IEEE, pp. 1–6, 2018, doi:10.1109/ICCUBEA.2018.8697406.

[5] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," in IEEE COMMUNICATIONS SURVEYS & TUTORIALS, vol. 21, no. 3, pp. 2702-2733, 2019, doi:10.1109/COMST.2019.2910750.

[6] V. Sharma et al., "Behavior and Vulnerability Assessment of Drones-Enabled Industrial Internet of Things (IIoT)," IEEE Access, vol. 6, no.1, pp. 43368–43383, 2018, doi: 10.1109/ACCESS.2018.2856368.

[7] G. George et al., "A Graph-Based Security Framework for Securing Industrial IoT Networks From Vulnerability Exploitations," IEEE Access, vol. 6, no. 1, pp. 43586–43601, 2018, doi:10.1109/ACCESS.2018.2863244.

[8] H. Sedjelmaci et al., "An Accurate Security Game for Low-Resource IoT Devices," IEEE Transactions on Vehicular Technology, vol. 66, no. 10, pp. 9381–9393,2017, doi: 10.1109/TVT.2017.2701551.

[9] H. Wu et al., "A Game Theory Based Collaborative Security Detection Method for Internet of Things Systems," IEEE Transactions on Information Forensics and Security, vol. 13, no. 6, pp. 1432–1445, 2018, doi: 10.1109/TIFS.2018.2790382.

[10] W. Meng, "Intrusion Detection in the Era of IoT: Building Trust via Traffic Filtering and Sampling," Computer, vol. 51, no. 7, pp. 36–43, 2018, doi: 10.1109/MC.2018.3011034.

[11] F. Yu et al., "Fast and memory-efficient regular expression matching for deep packet inspection," in Proceedings of the 2006 ACM/IEEE Symposium on

Architecture for networking and communications systems. ACM, pp. 93–102, 2006, doi: 10.1145/1185347.1185360

[12] L. Xiao et al., "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 41–49, 2018, doi:10.1109/MSP.2018.2825478.

[13] A. Saeed et al., "Intelligent intrusion detection in low-power IoTs," ACM Transactions on Internet Technology, vol. 16, no. 4, pp. 1-25, 2016, doi:10.1145/2990499.

[14] A. Abeshu et al., "Deep learning: the frontier for distributed attack detection in Fog-to-Things computing," IEEE Communications Magazine, vol. 56, no. 2, pp. 169–175, 2018, doi:10.1109/MCOM.2018.1700332.

[15] S. Sharmeen et al., "Malware Threats and Detection for Industrial Mobile-IoT Networks," IEEE access, vol. 6, pp. 941-957, 2018, doi:10.1109/ACCESS.2018.2815660.

[16] E. Hodo et al., "Threat analysis of IoT networks using artificial neural network intrusion detection system," in Networks, Computers and Communications (ISNCC), 2016 International Symposium on. IEEE, pp. 1–6, 2016, doi:10.1109/ISNCC.2016.7746067.

[17] T. Su et al., "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," IEEE Access, vol. 8, pp. 575 -585, 2020, doi:10.1109/ACCESS.2020.2972627.

[18] C. Yin et al., "A deep learning approach for intrusion detection using recurrent neural networks," Ieee Access, vol. 5, pp. 954-961, 2017, doi:10.1109/ACCESS.2017.2762418.

[19] Y. Ding et al., "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," in Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, 2018, pp. 81-85, doi:10.1145/3297156.3297230.

[20] C. Zhou et al., "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, no. 10, pp. 1345–1360, 2015, doi:10.1109/TSMC.2015.2415763.

[21] N. Muralidhar et al., "Illiad: InteLLigent Invariant and Anomaly Detection in Cyber-Physical Systems," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 9, no. 3, pp. 1–20, 2018, doi:10.1145/3066167.

[22] S. Prabavathy et al., "Design of cognitive fog computing for intrusion detection in Internet of Things," Journal of Communications and Networks, vol. 20, no. 3, pp. 291-298, 2018, doi:10.1109/JCN.2018.000041.

[23] A. Yassine et al., "IoT big data analytics for smart homes with fog and cloud computing," Future Generation Computer Systems, vol. 91, no. 1, pp. 563–573, 2019, doi:10.1016/j.future.2018.08.040

[24] R. Kumar et al., "A Survey: Review of Cloud IoT Security Techniques, Issues and Challenges," in International Conference on Advanced Computing and Software Engineering (ICACSE-2019, pp.447-452, 2019, doi:10.2139/ssrn.3350995

[25] E. Viegas et al., "A resilient stream learning intrusion detection mechanism for real-time analysis of network traffic," in 2017 IEEE Global Communications Conference(GLOBECOM 2017), IEEE, pp. 1–6, 2017, doi:10.1109/GLOCOM.2017.8254495.

[26] P. Mulinka et al., "Stream-based machine learning for network security and anomaly detection," in Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks. ACM, pp. 1–7, 2018, doi:10.1145/3229607.3229612.

[27] A. Divekar et al., "Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives," in 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). IEEE, 2018, pp.1-8, doi:10.1109/CCCS.2018.8586840.

[28] M. Tavallaee et al., "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE, 2009, pp. 1-6, doi:10.1109/CISDA.2009.5356528.

[29] A. Shiravi et al., "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," computers & security, vol. 31, no. 3, pp. 357–374, 2012, doi:10.1016/j.cose.2011.12.012.

[30] I. Sharafaldin et al., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *International Conference on Information Systems Security and Privacy* (ICISSP), pp. 108–116, 2018, doi:10.5220/0006639801080116

[31] A. H. Lashkari et al., "Characterization of Tor Traffic using Time based Features," in *International Conference on Information Systems Security and Privacy* (ICISSP), pp. 253–262, 2017, doi:10.5220/0006105602530262.

[32] N. Koroniotis et al., "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, no. 1, pp. 779–796, 2019, doi:10.1016/j.future.2019.05.041.

[33] N. Moustafa et al., "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 military communications and information systems conference (MilCIS). IEEE, pp. 1–6, 2015, doi:10.1109/MilCIS.2015.7348942.

[34] Y. Meidan et al., "N-BaIoTNetwork-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018, doi:10.1109/MPRV.2018.03367731.

[35] J. Karande et al., "NSL-KDD in JSON Format," IEEE Dataport, 2020. doi:10.21227/w3f0-nr98.

[36] J. Karande et al., "NSL-KDD in AVRO Format," IEEE Dataport, 2020. doi:10.21227/134f-7171.

B. Mishra and A. Kertesz, "The Use of MQTT in M2M and IoT Systems: A Survey," in IEEE Access, vol. 8, pp. 201071-201086, 2020, doi:10.1109/ACCESS.2020.3035849.