Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 8, August : 2022

SECURE DATA TRANSMISSION IN THE IOT CLOUD ENVIRONMENT: AN OPTIMIZATION ALGORITHM BASED ON AI

[1]D. Lakshmi Narayana Reddy
[1]Assistant Professor
Anantha Lakshmi Institute of Technology & Sciences , Anantapuramu, A. P, India.
Department of Computer Science and Engineering
Mail id: lakshmi1217@gmail.com.

Due to its inherent latency, Traditional Cloud Computing is not a viable option for safely storing data from the Internet of Things (IoT), particularly considering the ever-increasing quantity of IoT sensors and physical devices connected to the Internet. Since processing such massive amounts of data on Cloud facilities is not an easy task, there is a dearth of literature on the topic of automating security measures for all parts of the IoT-Cloud ecosystem that deal with real-time activities and massive amounts of data. In the context of managing many devices located in different locations, the process of creating a secure and automated data transfer from the IoT layer to the cloud layer may be rather challenging. An advanced technique for strengthening security protocols inside an IoT cloud environment is introduced in this chapter. For this particular objective, the butterfly optimization method is used.

1. Introduction

Both cloud computing and the Internet of Things (IoT) have grown in popularity in the last few years. Numerous Internet of Things (IoT) devices are now available, each one tailored to a certain set of user requirements. Also, top cloud service providers are adding an extensive array of Internet of Things (IoT) capabilities to their software offerings. In line with the growth of this emerging phenomena, there has been a lot of focus on studying the security of smart IoT cloud systems in the last few years. Nowadays, the concept of a network of interconnected devices known as the Internet of Things (IoT) has become more popular. With the use of networking technology, people may get data from a variety of sources and manipulate it in various ways, which improves their interaction with the world around them. Hardware and networking technology have come a long way in the last decade, which may explain why IoT devices are so popular. In addition, the GSMA estimates that there will be

25.2 billion devices in use globally by 2025 as a consequence of the continued deployment of Internet of Things (IoT) devices [8]. Nowadays, cloud computing has emerged as a distinct technical framework inside contemporary society, living alongside the Internet of Things (IoT). Device type, network connection, time restrictions, and geographical location are not barriers to service use.

The storage, communications, and processing backend capabilities of cloud infrastructure may be used by IoT devices. This paves the way for the deployment of IoT terminal apps that can access data remotely and make use of computing capabilities. Although cloud computing and the Internet of Things (IoT) have evolved separately, academics have begun to combine the two in order to improve the performance and usefulness of IoT applications. Many well-known cloud providers are now offering IoT-related cloud services to their customers. It is becoming increasingly clear that technical advances in the IoT cloud market are having a significant impact. Echo by Amazon [18] and its services are a model of a robust IoT cloud ecosystem. This approach involves collecting speech data from customers and sending it to the cloud via the Alexa device's microphones. The cloud will then provide Alexa the results after the data processing is complete. For users with a wide variety of Internet of Things (IoT) devices installed throughout their homes, Alexa may act as a central hub for all of their networking needs. This includes things like getting food, showing a picture, and turning on the TV. If you want to talk to Alexa from afar while you're not at home, one option is to use a terminal app, such one for your phone. To accomplish these goals, cloud services are used.

Several industries are seeing a rise in the use of Internet of Things (IoT) cloud ecosystems, such as smart homes, autonomous vehicles, healthcare, and industrial machinery. Concerns about safety and security, however, remain paramount. Recent studies have focused on the integration of IoT with cloud computing. Since some IoT cloud systems connect to critical infrastructures and all of them to people, it is essential to understand the security protocols used by these systems. Protecting these systems and helping to create more successful techniques requires a complete understanding of them and the people who use them. This chapter's goal is to provide a comprehensive review of the current state of knowledge and future research roadblocks related to consumer-centric IoT cloud platforms. Anybody interested in this area of study, from practitioners to academics, may use this resource as a reference. In order to encourage the creation of better solutions for IoT cloud systems, this chapter seeks to address the current research concerns.

2. Literature review

The many facets of cloud computing and the Internet of Things (IoT) security have been the subject of a great deal of scholarly investigation. With a focus on studies and worries about IoT security, Sicari et al. [11] investigated the subject of security in IoT systems. Additionally, this article evaluated the current Internet of Things (IoT) applications. Hardware, software, and networking are all parts of the Internet of Things (IoT) that might be vulnerable to attacks, according to Alaba et al. (2019). For the purpose of enhancing security within the IoT domain, Khan and Salah conducted research to investigate potential blockchain-based solutions. In their investigation of Internet of Things (IoT) security threats and requirements, Harbi et al. The field of Internet of Things data forensics was thoroughly investigated by Stoyanova, who looked at several facets, including problems, theoretical frameworks, and practical solutions. Cloud computing services pose security issues that need attention and have potential solutions, say Khalil et al. Additionally, the writers looked at present problems and potential solutions concerning cloud computing security [14, 10]. Domingo-Ferrer et al. performed an evaluation that looked at several privacy-protecting strategies for data stored in the cloud. The purpose of the survey by Ahmed et al. (year) [3] was to evaluate trust in the setting of cross-cloud federation. Regarding the privacy and security issues of cloud computing, Tabrizchi (2011) conducted an extensive study. Examining the architectural and security components of the IoT cloud, Ammar et al. performed a study of its integration. The citation given as reference [9] documents this discovery. Dizdarevic et al. (2019) investigated the protocols for data exchange between the cloud, fog computing, and the Internet of Things (IoT). On the other hand, Celik et al. (2017) used program analysis techniques to investigate the privacy and security flaws in IoT programming platforms. Researchers Kumar et al. evaluated security procedures and conducted a thorough risk assessment for IoT applications hosted in the cloud. When it comes to Internet of Things (IoT) applications hosted in the cloud, Almolhis analyzed the main security challenges and existing solutions. Internet of Things (IoT) cloud ecosystems, which bring together IoT devices with cloud computing, have showed promise in earlier research as a means to improve consumer-facing intelligent applications. Internet of Things

(IoT) cloud integration is a relatively new feature in consumer applications, while it has been around for a long time in academia. It is common for consumer applications to have a big user base, often exceeding one million. Researchers in the academic sector are now looking at ways to better safeguard their interests. There is a new security risk, and past evaluations didn't do enough to deal with it. The reasoning for this is because a single Internet of Things (IoT) system or cloud application cannot compare to the size of a typical IoT cloud ecosystem. Smart home technology, including voice-activated assistants, have been integrated into many people's homes. There may be significant obstacles to overcome in order to ensure the security of data while efficiently managing a massive number of devices. An IoT cloud environment opens the door to additional potential security breaches due to the greater ease of access it provides.

Any individual may theoretically get the necessary hardware and software to connect to the Internet of Things (IoT). For the part of the cloud that matters, public HTTP GET/PUT services make it easier for Internet of Things (IoT) devices to communicate with the cloud. Also, devices from different manufacturers, owned by different people, and incorporated into the IoT cloud's architecture may all be linked via an IoT hub. The wide variety of entry points used by devices and people increases the difficulty of maintaining system security. With the recent completion of its third expansion, the IoT cloud ecosystem now offers an even wider variety of services and solutions. Commercially available IoT cloud applications are often used by several distinct customers, with each customer purchasing a separate device of the same sort.

Consequently, the cloud endpoint has the difficult and time-consuming duty of differentiating between several users. If two users share the same brand of Internet of Things (IoT) equipment and one of them makes it clear that they don't want the other user to be able to access their data, then... In addition, the ecosystem that includes the cloud for the Internet of Things has a higher degree of human involvement. In addition, the platform offers a mobile app that may collect data on human activities, similar to smart home apps, and help with the administration of IoT devices. Every individual has their own distinct habits and tastes, and it's possible that a single device might be used and accessible by many people. An additional risk of physical attack increases with the level of human involvement. It was challenging to comprehend and use real, working Internet of Things (IoT) systems at an intuitive and practical level due to the previous evaluation's lack of concrete examples, as shown in the references mentioned [4, 9, 14, 16, 4, 7].

3. Proposed Model

First, we'll look at a simple example of a smart home application to help clarify the concept. Given this history, it's reasonable to say that the smart air conditioner and the temperature sensor are cornerstones of the IoT architecture. A smart home hub will begin sending temperature readings to a server in the cloud the moment the temperature exceeds 30 degrees. The data analysis system on the cloud server has found that the present temperature is higher than what is considered acceptable. An order from a remote server in the cloud triggers the smart air conditioner to turn on and set the desired temperature.

The smart air conditioner's primary function is to set and maintain the temperature you want. Even when the outside temperature is 30 degrees Celsius, a sick person may still be reluctant to turn on the air conditioner. Without the user's explicit permission, using a smartphone app or other control interface to activate the intelligent air conditioning system may be hindered. While it's impressive that an IoT cloud ecosystem might potentially act as a self-adaptive system, it's crucial to remember that it's still susceptible to human interference. An app that caters to the needs and tastes of its users is the main object of this review. Everyone from customers to manufacturers to cloud service providers to society at large is worried about the security of new apps. These applications are now being used by a large number of people. An objective assessment of the security mechanisms used by forthcoming consumer-facing Internet of Things (IoT) cloud platforms would be beneficial

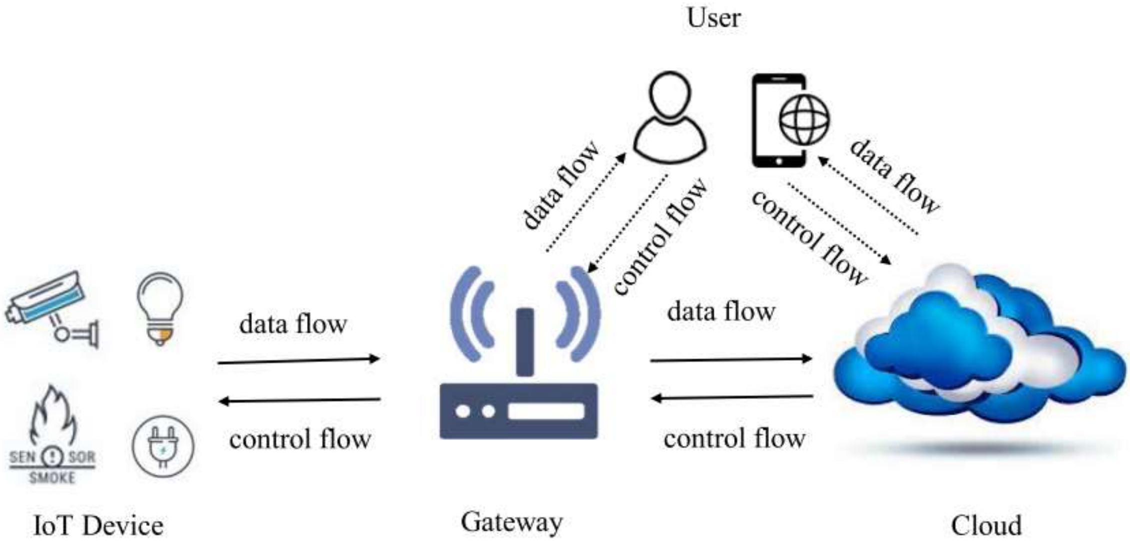


Figure 1: IoT Cloud ecosystem

The act of ensuring the security of the gateway has significant relevance within the Internet of Things (IoT) cloud ecosystem, as seen in Figure 4.1. This chapter presents a sophisticated technique that involves the extraction and categorization of attack aspects at the gateway, within a specified context.

3.1 Feature selection using bat induced butterfly optimization (BBO)

Feature selection is a preliminary methodology used to improve the overall quality of a product. The idea of feature selection (FS) encompasses a range of optimization strategies that aim to determine the most optimum subset of attributes within a given database. The primary purpose of FS is to properly replicate the original data. The feature selection method typically consists of two main stages: the initial identification of the minimal reduction and the subsequent evaluation of the selected features. The fundamental challenge is in assessing the continued viability of the optimum feature selection approach with respect to the attributes of the source data. Fortunately, FS is considered a search entity that encompasses a portion of the characteristics at each point in the search area. The present work used a bat-induced butterfly optimization (BBO) technique to discern the most suitable characteristics and minimize superfluous data.

The first change is that we use a certain frequency and sound instead of a different frequency g_j. In BBO, each bat is determined by its position y_j^T , velocity U_j^T . The new solutions y_j^T and velocities U_j^T at time step T are given by

$$U_j^T = U_j^{T-1} + (y_j^T - y_*)g \tag{1}$$

$$y_j^T = y_j^{T-1} + U_j^T \tag{2}$$

The global best solution is referred as y∗. In this g is equal to 0.5. To increase demographic diversity the search performance is improved by Eq. (3)

$$Y_{NEW} = y_{s1}^T + G(Y_{s2}^T - Y_{s3}^T) \tag{3}$$

where G is the mutation weight factor, while s_1 , s_2 , s_3 are evenly divided into random numbers. The migration process can be expressed as follows:

$$y_{j,z}^{T+1} = y_{s1,z}^{T} \quad (4)$$

where $y_{j,z}^{T+1}$ zth denotes an element of y_j at generation T+1 it gives the position of King Butterfly *i*. Similarly, $y_{s1,z}^{T}$ indicates the *z*th newly formed stage of the monarch butterfly s_1. *T* is the number of the current generation. Monarch butterfly s_1 is approximately selected from the sub-population. Here, *s* can be calculated as

$$s = Rand * Peri \quad (5)$$

Peri indicates immigration period. Rand is a random number obtained as a result of consolidated distribution. Or rather, if s>q, the kth element in the butterfly is the newly formed king

$$y_{j,z}^{T+1} = y_{s2,z}^{T} \quad (6)$$

where $y_{j,z}^{T+1}$ the newly formed phase of the monarch butterfly is the return element s_2. Monarch butterfly r_2 is approximately selected from the sub-population. If the generated probable number q is less than or equal to q for all components of the monarch butterfly, it can be updated as follows:

$$y_{j,z}^{T+1} = y_{Best,z}^{T} \quad (7)$$

where $y_{j,z}^{T+1}$ zth denotes an element of y_i at generation T+1 gives the position of King Butterfly j. Similarly, $y_{Best,z}^{T}$ zth denotes an element of y_{Best} that is Best King Butterfly in Land 1 and Land 2. T is the number of the current generation. Or rather, if larger than the Rand P, it can be upgraded

$$y_{j,z}^{T+1} = y_{s3,z}^{T} \quad (8)$$

where $y^{T}_{s3,z}$ and zth denotes an element of y_{s3}. In this case, if it is Rand >BAR, it can be updated as follows

$$y^{T+1}_{i,z} = y^{T+1}_{i,z} + \alpha \times (dy_z - 0.5) \qquad (9)$$

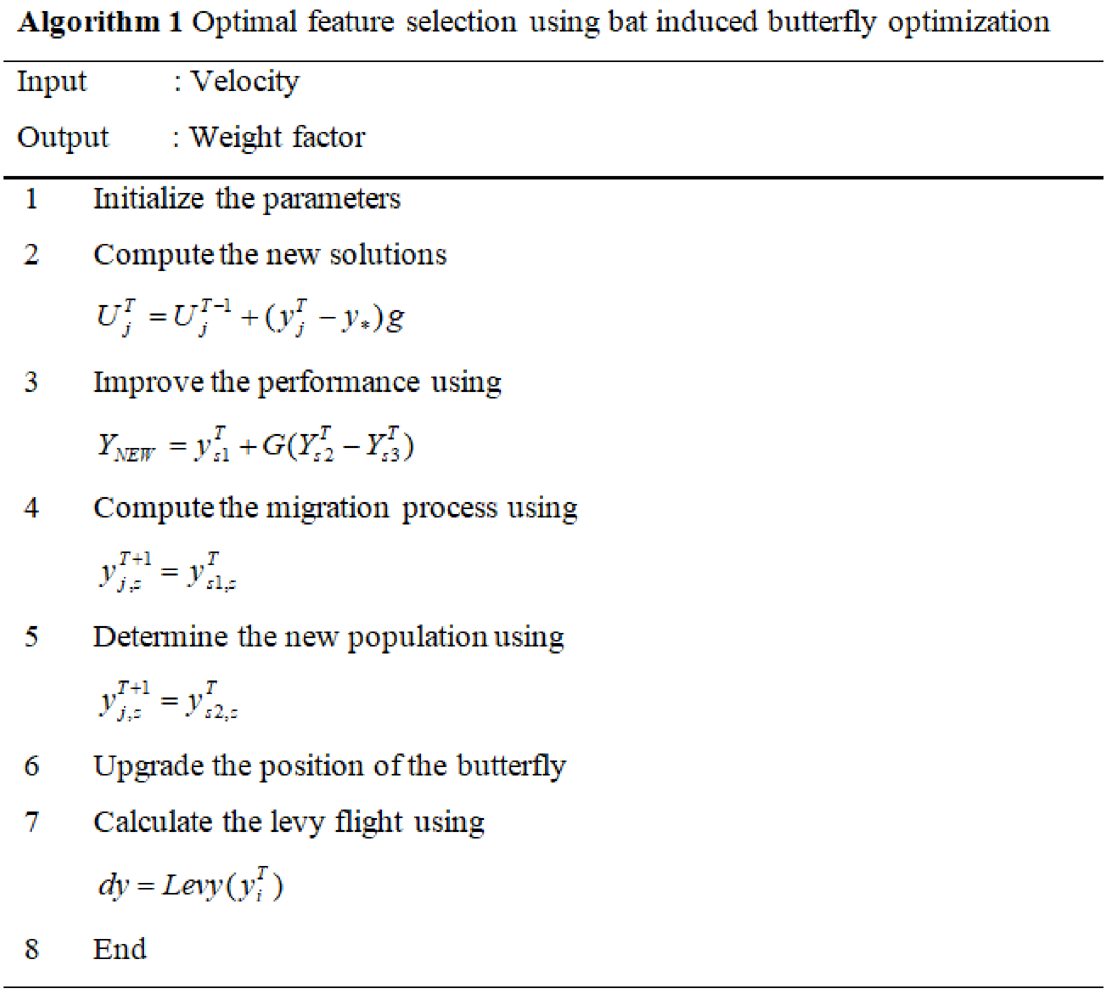
where it indicates butterfly adjustment speed. dy is the according to the monarch butterfly i Levy calculate this by flight.

$$dy = Levy(y^{T}_{i}) \qquad (10)$$

In Eq. (9), α is the expectation factor is given as Eq. (11)

$$\alpha = R_{Max} / T^{2} \qquad (11)$$

The working function of algorithm 1 represents the function of the BBO.



Algorithm 1 Optimal feature selection using bat induced butterfly optimization

```
Input    : Velocity
Output   : Weight factor

Initialize the parameters
Compute the new solutions
    $U_j^T = U_j^{T-1} + (y_j^T - y_*)g$
Improve the performance using
    $Y_{NEW} = y_{s1}^T + G(Y_{s2}^T - Y_{s3}^T)$
Compute the migration process using
    $y_{j,z}^{T+1} = y_{s1,z}^T$
Determine the new population using
    $y_{j,z}^{T+1} = y_{s2,z}^T$
Upgrade the position of the butterfly
Calculate the levy flight using
    $dy = Levy(y_i^T)$
End
```

3.2 Classification using Random Forest algorithm

Bremen is widely acknowledged as the originator of the first random forest algorithm, with its development dating back to 2001. The use of decision trees is utilized by a classification methodology to manage the random forest. Data mining methods, such as the decision tree algorithm, are extensively used in several domains. The categorization process in decision trees entails the use of both current data and data qualities to produce a well-informed determination about the class or category. The Classification and Regression Tree (CART) algorithm is a fundamental component of the decision tree approach, characterized by its binary tree structure. Based on the reference provided (33), it can be seen that each stage of the random forest comprises four CART trees. During the training phase, a subset of training samples is picked using the Bootstrap sampling technique. This subset is denoted as D1, D2,..., Dk. In the end, the decision tree labeled as K will be formed. Based on the criterion of minimum purity, it is recommended to choose the most outstanding expert only from the set of candidate M branches at node N inside the classification tree. As a result, the trees will attain full growth. The third step involves a replication of the previous phase. The user's text is too short to be rewritten in an academic manner. A decision tree was created. The formation of the asymmetrical forest is attributed to the presence of key trees that possess a firmly established presence. The ongoing refinement of the final sample selection procedure in the random forest methodology necessitates more advancements.

The assessment of all the characteristics is performed using a multi-class support vector machine (SVM) classifier for each unique combination of features. The classification is conducted by using an initial dataset, in which only the relevant qualities for the given job are included. Afterwards, the hypothesis is tested by using filtered experimental data sets. The implementation of a one-vs-all strategy is considered essential in order to provide separate classifications for each group. Ultimately, the evaluation of the feature subset is conducted by analyzing its classification performance on the experimental data via the use of various support vector machines. The process of encoding attributes entails the employment of binary strings that are representative of the quantity of characteristics. In the present encoding technique, the value of a binary digit of zero indicates the non-selection of a specific attribute, while a binary digit of one indicates the selection of the attribute as a constituent of an attribute subset.

The algorithm under consideration is a meta-heuristic that combines innovative approaches like local search with traditional search methods such as evolutionary algorithms. The memetic algorithm is a computational methodology that integrates components of evolutionary algorithms and local search techniques in order to optimize solutions for intricate issues. Improve the effectiveness of the fundamental search algorithm by minimizing the time needed to get an ideal outcome [22]. Evolutionary algorithms are often formulated to explore the whole range of the search space. In contrast, a localized exploration inside a defined geographic region use an evolutionary process to uncover more optimal solutions. The performance of an algorithm is greatly influenced by the choice of generation operators, as well as the program's categorization and local search methodology. The current research utilizes a local search methodology to assess the closeness of the answer after it is received by the distribution estimation algorithm. The method employs a selection process to determine the subset that is both practicable and closest in proximity among the given possibilities, with the aim of determining the most optimum choice. In the end, the most favorable option is replaced with the existing one.

4. Performance Analysis

4.1 Data Set

The NSL-KDD dataset consists of records that have 43 fields. Attribute 41 is linked to a closed behavior field that represents the distinctive behavior or intrusion type. The last field pertains to the level of challenge involved in detecting the intrusion. The column denoted as "label" has five distinct classifications, including a solitary category for conventional attacks and three categories for intrusions: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probable (Prob). A denial of service attack, often known as a DoS attack, is a kind of cyber assault in which a targeted computer system is overwhelmed with an excessive number of connection requests, resulting in the system's inability to meet legitimate user demands. Consequently, the server incurs substantial financial obligations and becomes inept in efficiently managing typical network traffic. The process of targeting the root account entails using a normal user account to get higher access by exploiting a vulnerability inherent within the system. During an external intrusion attempt, the perpetrator has the capacity to deliver packets to a computer system. However, the perpetrator lacks an authentication credential on the targeted device, hence hindering their ability to access the system in a manner similar to that of an authorized user. During the process of intrusive scanning infiltration, the system undergoes a thorough scan with the objective of finding prospective vulnerabilities or attacks that may be prone to exploitation in future occurrences. The identified vulnerabilities possess the capacity to be used for the purpose of executing a system attack. The existing dataset classifies numerical and textual data into three distinct categories, namely basic, content, and traffic.

The TCP/IP capabilities serve as the fundamental elements of an IP connection. The presence of these attributes hinders the prompt identification of security breaches. When examining a network connection, several factors are taken into account, such as the duration of the connection, the protocol and service used, and the volume of data sent during the connection. The paragraph examines the characteristics or attributes of information. The observed attacks exhibit a distinct lack of consistent aberrant repetition, hence distinguishing them from other forms of assaults that seek to interrupt services and engage in scanning operations. The root system's vulnerability stems from its constrained connectivity, whereby network data packets are confined to the data part and hold a solitary link. This is in opposition to service-blocking and scanning assaults, which create several connections to servers within a short timeframe. The inclusion of packet data analysis services is of utmost importance in order to identify indications of infiltration activities, such as the frequency of failed attempts. The identification of this specific kind of attack is crucial. The phrase "content attributes" is often used to refer to these particular features. Instances of these attributes include the regularity of network engagements, the quantity of failed login endeavors on a network, and the user's authorization to enter the system in the capacity of an administrator. In the context of traffic characteristics, it is apparent that they may be classified into two distinct categories. There are two types of relationships that may be classified based on temporal factors. The first group pertains to connections that demonstrate both same service and host as the current connection, transpiring within a timeframe of two seconds. The second classification of temporal connection is used for the purpose of investigating protracted acts of aggression. The aforementioned qualities may be classified as machine-based, since they quantify the ratio of past connections to present connections that possess identical service and host parameters. The methods are subjected to cross-validation using the CIDDS-001, KDD99, and VIRUS TOTAL datasets.

4.2 Simulation Results

This section of the study article presents the results collected from the NSL-KDD database. This study investigates the efficacy of five distinct feature selection methodologies via the use of the support vector machine algorithm on populations including 50, 100, and 150 people. The algorithms of leading selection and backward selection are considered to be population-independent, since their performance is unaffected by an increase in population size. In the context of smaller populations, the method under evaluation shown superior performance when compared to the distribution estimation technique. However, as populations have increased, the level of accuracy between the two methods has diminished. The use of the distribution estimate strategy in conjunction with local search techniques has resulted in significant improvements in the performance of small populations. The consistency of the postulated mechanism's correctness is seen across all levels of the population, as shown in Figure 2.

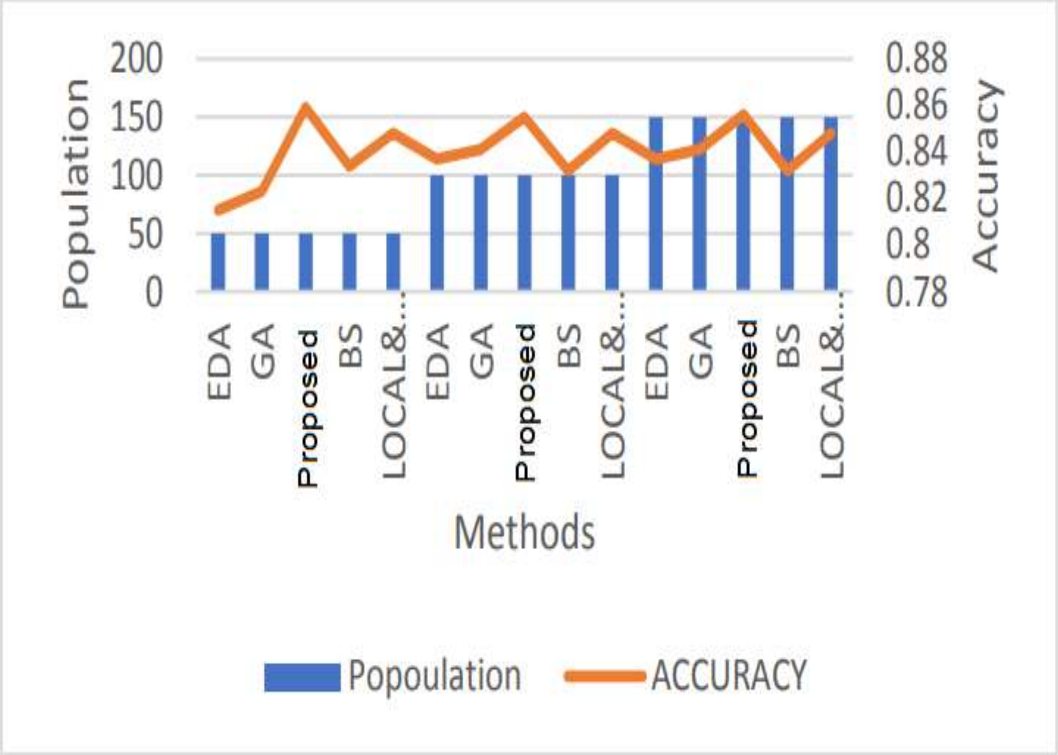


Figure 2: Accuracy evaluation with other algorithms

The diagram shown above illustrates the classification accuracy of a database consisting of packets that have been categorized into five unique groups. The study demonstrates the level of precision achieved by using various feature selection methods across populations of different sizes. The detection accuracy of effects is notably diminished when the training database comprises a restricted number of samples, resulting in a decline in the overall accuracy of detection.

5. Conclusion

The process of identifying and classifying data is of great importance in detecting unauthorized access, with the thoughtful selection of relevant qualities being a crucial determinant. Feature selection approaches may be used on large datasets to enhance the performance of classifiers, while also lowering the time and cost associated with detection. The objective of this study was to evaluate the efficacy of several genetic attribute selection procedures, distribution estimation, hybrid distribution estimation with local search, leading selection, and backward selection in SVM classification, taking into account their algorithmic suitability. Hence, it was essential to do a comprehensive investigation to evaluate the influence of four feature evaluation metrics on the classification accuracy of an Intrusion Detection System (IDS). The achievement was attained by the implementation of a thorough experiment that included two well acknowledged benchmark datasets, namely NSL-KDD and VIRUS TOTAL, as well as four advanced machine learning classifiers, namely KNN-RF, PSO, SVM GA, and GA. The results obtained from all classifiers had a similar character. However, the proposed technique exhibited the highest level of precision in detecting, as shown by several feature evaluation metrics, when the optimal parameter values were used.

6. References

[1] Anhtuan Le, Jonathan Loo, Yuan Luo, and A. Lasebae. The impacts of internal threats towards routing protocol for low power and lossy network performance. pages 000789–000794. 2013. ISBN 978-1-4799-3755-4. doi: 10:1109/ISCC:2013:6755045.

[2] Ahmet Arıs¸, Sıddıka Yalc¸ın, and Sema Oktug. New lightweight mitigation techniques for rpl version number attacks. Ad Hoc Networks, 85, 2018. doi: 10:1016/j:adhoc:2018:10:022.

[3] Amit Dvir, Tam´as Holczer, and Levente Butty´an. Vera - version number and rank authentication in rpl. pages 709–714. 2011. doi:10:1109/MASS:2011:76.

[4] Heiner Perrey, Martin Landsmann, Osman Ugus, Thomas Schmidt, and Matthias W¨ahlisch. Trail: Topology authentication in rpl. 2013.

[5] Ghada Glissa, Abderrezak Rachedi, and Aref Meddeb. A secure routing protocol based on rpl for internet of things. 2016. doi:10:1109/GLOCOM:2016:7841543.

[6] Anthea Mayzaud, Remi Badonnel, and I. Chrisment. A distributed monitoring strategy for detecting version number attacks in rpl-based networks (invited paper). IEEE Transactions on Network and Service Management, PP:1–1, 2017. doi:10:1109/TNSM:2017:2705290.

[7] David Airehrour, Jairo A. Gutierrez, and Sayan Kumar Ray. Sectrust-rpl: A secure trust-aware rpl routing protocol for internet of things. Future Generation Computer Systems, 93:860 – 876, 2019. ISSN 0167-739X. doi:https://doi:org/ 10:1016/j:future:2018:03:021.

[8] Joydeep Tripathi, Jaudelice De Oliveira, and Jp Vasseur. A performance evaluation study of rpl: Routing protocol for low power and lossy networks. pages 1–6. 2010.

[9] Sniderman, B.; Mahto, M.; Cotteleer, M.J. Industry 4.0 and Manufacturing Ecosystems; Deloitte University Press: London, UK, 2016; pp. 1–23.

[10] Corotinschi, G.; G ˘aitan, V.G. Enabling IoT connectivity for Modbus networks by using IoT edge gateways. In Proceedings of the 2018 International Conference on Development and Application Systems (DAS), Suceava, Romania, 24–26 May 2018; pp. 175–179.

[11] Geissbauer, R.; Schrauf, S.K.V. Industry 4.0-Opportunities and Challanges of the Industrial Internet. Available online: https: //www.strategyand.pwc.com/gx/en/insights/2015/industrial-internet.html (accessed on 2 February 2021).

[12] Frankó, A.; Vida, G.; Varga, P. Reliable Identification Schemes for Asset and Production Tracking in Industry 4.0. Sensors 2020, 20, 3709. [CrossRef]

[13] Massaro, A.; Galiano, A. Re-engineering process in a food factory: An overview of technologies and approaches for the design of pasta production processes. Prod. Manuf. Res. 2020, 8, 80–100. [CrossRef]

[14] Weerasiri, D.; Barukh, M.C.; Benatallah, B.; Sheng, Q.Z.; Ranjan, R. A Taxonomy and Survey of Cloud Resource Orchestration Techniques. ACM Comput. Surv. 2017, 50, 1–41. [CrossRef]

[15] Maiti, P.; Shukla, J.; Sahoo, B.; Turuk, A.K. QoS-aware fog nodes placement. In Proceedings of the 2018 4th International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 15–17 March 2018; pp. 1–6. [CrossRef]

[16] Groover, M. Fundamentals of Modern Manufacturing: Materials, Processes, and Systems; John Wiley & Sons, Inc: Hoboken, NJ, USA, 2020.

[17] Deshmukh, U.; More, S.A. Fog Computing: New Approach in the World of Cloud Computing. FInt. J. Innov. Res. Comput.Commun. Eng. 2016, 4, 16310–16316. [CrossRef]

[18] Luan, T.H.; Gao, L.; Li, Z.; Xiang, Y.; Wei, G.; Sun, L. Fog computing: Focusing on mobile users at the edge. arXiv 2015, arXiv:1502.01815

[19] Puliafito, C.; Vallati, C.; Mingozzi, E.; Merlino, G.; Longo, F.; Puliafito, A. Container Migration in the Fog: A Performance Evaluation. Sensors 2019, 19, 1488. [CrossRef]

[20] Gil, D.; Ferrández, A.; Mora-Mora, H.; Peral, J. Internet of things: A review of surveys based on context aware intelligent services. Sensors 2016, 16, 1069. [CrossRef]

[21] Perera, C.; Qin, Y.; Estrella, J.C.; Reiff-Marganiec, S.; Vasilakos, A.V. Fog Computing for Sustainable Smart Cities. ACM Comput. Surv. 2017, 50, 1–44. [CrossRef]

[22] Naha, R.K.; Garg, S.; Georgakopoulos, D.; Jayaraman, P.P.; Gao, L.; Xiang, Y.; Ranjan, R. Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions. IEEE Access 2018, 4, 1–31. [CrossRef]

[23] Maag, B.; Zhou, Z.; Thiele, L. A survey on sensor calibration in air pollution monitoring deployments. IEEE Internet Things J. 2018, 5, 1–15. [CrossRef]

[24] Mukherjee, M.; Shu, L.; Wang, D. Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Commun. Surv. Tutor. 2018, 20, 1–30. [CrossRef]

[25] Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 2015, 17, 2347–2376. [CrossRef]

[26] Yassein, M.B.; Shatnawi, M.Q.; Aljwarneh, S.; Al-Hatmi, R. Internet of Things: Survey and open issues of MQTT protocol. In Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS), Monastir, Tunisia, 8–10 May 2017. [CrossRef]

[27] Maheswari, K.; Bhanu, S.S.; Nickolas, S. A Survey on Data Integrity Checking and Enhancing Security for Cloud to FogComputing. In Proceedings of the IEEE Xplore, Bangalore, India, 5–7 March 2020; pp. 121–127.