INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 8, Issue 3 - V8I3-1395)
Available online at: https://www.ijariit.com

Development of lightweight and secure IoT device management framework

Faith Ebosetale Omohodion
omohodion.faith10@gmail.com
Gandhi Institute of Technology and Management, Visakhapatnam, Andhra Pradesh

S. Padma
psurakas@gitam.edu
Gandhi Institute of Technology and Management, Visakhapatnam, Andhra Pradesh

ABSTRACT

The Internet of Things (IoT) is quickly expanding, posing serious security problems across the network. A security framework is required due to the enormous amount of communication and the fact that it operates in real time. The Internet of Things (IoT) paradigm necessitates ubiquitous connectivity to billions of disparate devices. Internet of Things (IoT) devices has extremely low processor, memory, and storage capabilities. Therefore, a lightweight security process is to be provided for IoT devices in other to create required development. This paper introduces several development in lightweight versions of various known security processes, analyzes each relevant process. With a machine-learning-based model process that provides a lightweight solution that works by selecting the best features leveraging well-known filter and wrapper methods for feature selection. The strategy can be evaluated over different datasets collected from varying network scenarios.

Keywords: *Internet of Things, Development, Security, Lightweight, Device Management.*

1. INTRODUCTION

Devices and items with built-in detectors are connected to an Internet of Things platform, which combines data fromvarious devices and applies analytics to share the most useful information with apps tailored to individual needs.The Internet of Things is the idea of linking any gadget to the Internet and other network devices (as far as it has an on/off switch). Device management framework is an essential component of any IoT platform and security is a very imperative part of IoT so considering the IoT security measures, exceptional requirements are desired to fulfill the needs of IoT.

In recent developments of secured lightweight process strategy of a sampling technique and machine learning technique that uses Random Forest performance in utilizing IoT devices and then the use of classification method of sampling data to provide the best way to secure IoT device management has been made.

2. SECURITY REQUIREMENTS IN AN IOT

A. Data confidentiality: Only two legitimate end points, that is, the sender and the receiver in a network can read the data transmitted. In IoT-based IA, the sensor nodes transmit data collected for each item to the coordinating unit (CU). The CU passes this data to the server for further decision making. Any intruder will be unable to read this inventory status information due to data secrecy.

B. Data integrity: Sensor nodes are responsible for collecting and forwarding the data related to the item availabilityto the coordinator. Any intruder can gain access to this data or attempt to modify them for any criminal purpose, such as filing fake reports or stealing items. Therefore, in an IoT-based IA system, it becomes the prime responsibility to protect the information from the disclosure.

C. Authentication: Confirming one's identity to another is crucial for security operation of IoT. The sensor nodes near each item relay information to the central CU in AI. In turn, the CU sends the data to the server. Mutual authentication between the central controlling unit as well as the server is required before data exchange can begin. This will aid in recognizing and proving each other's identities.

D. Data validation: The data received by the controlling unit must be fresh. The intruder, on the other hand, may retrieve your validated information and transmit it to CU using the old key after some time. This may refrain the CU from receiving the correct status in the formation of an item in inventory.

E. Availability: For smooth working of the IoT and access to data whenever needed, it is also important that services that applications offer should be always available and work properly. Intrusions and hostile activity, in other words, must be identified. In addition to the security techniques listed above, intrusion detection systems (IDSs) and firewalls are employed to assure the availability of security services.

Even though standard security will solve many issues, there are some unique elements of IoT security that demand a comprehensive and integrated approach to IoT security. IoT devices are frequently deployed in an unsupervised method, making physical attacks more likely. Due to their limited resources, low-end IoT devices are unable to run more complex cryptographic algorithms. Wireless communication is used by a huge number of IoT devices, whichis susceptible to hacking and jamming. This study explores ways to safely link the IoT into the Internet in attempt to improve IoT security, and makes many key contributions:

1. We propose a self-contained architecture that enables for variable contour adaption to the nature of the domain.
2. The goal is to propose a flexible way to compactly represent the data with multiple parameters that may be chosenand altered, rather than to compare the performance of yet another classifier.
3. Lastly, we strive for the best sampling strategy given sequential data, generated from IoT devices.

The current focus of Internet of Things (IoT) research is on improving the security of every message sent across the network. Keeping this thought in mind, researchers in this work emphasizes a security oriented cryptographic solution. The key size, operations, and technique that commonly used security cryptography systems use to secure a message are all quite large. This research first examines the advantages of using lightweight security cryptography methods in the Internet of Things as lightweight does not intend to be weak in nature; The attacker is limited by the lightweight techniques, which expose only limited data controlled by a single key. For maintaining a tradeoff between performance, security, and resources required, lightweight solutions are used

3. ABOUT THE PROPOSED TECHIQUES AND DATA EXECUTION

To first predict the behavior of a system, past data are first examined to discover common patterns and other classification issues in order to see algorithms or tools suitable for improvement. In this research, we present a method and a system that uses far fewer resources while maintaining high forecast accuracy and capability. A common behavior graph, the contour enclosing the graph, and entropy calculation methods are the three fundamental methodologies used. We then show how it is applicable for IoT device management, using machine learning algorithms.

Sampling Technique

We analyze sampling data collected over several periods in this research we will be using banking data of online payment, and then divide the period into time-units. We can divide a year into daily time units, for example. We obtain one value that represents each time-unit. This is accomplished by averaging the samples obtained over the course of the time unit. We may calculate the average value of all samples from that day in the example. We might alternatively choose one of the samples to symbolize the day, such as the first or last. The mean values for each time-unit is then calculated from the collected values for the same time-unit throughout all periods, yielding an overall average for that time-unit. We repeat this process for all time units in the period, resulting in a graph that depicts the average values for a typical and average period.

We now calculate the contour around the average graph line for an average time, assuming we have the average graph line. The resulting contour depicts the standard range of values, and can be used to compare an unanalyzed period to it. The interval is a conventional period if its graph value is totally within the contour. It is fully non- standard if it is entirely out of the contour. If certain sections of the graph are within the contour but others are not, we calculate the overall "distance" of the given time from the conventional contour using an entropy measurement. We can determine whether the period is standard or not if there is an existing entropy threshold. At the unit level, weuse the same principle to determine if a certain time-unit in a period is within the standard or not. This particular check is useful for detecting anomalous IoT activities, for example.

Finally, the entire procedure is built on three crucial elements: the average graph per period, the contour around the mean graph, and an entropy value that represents a period's overall distance from the contour. Every one of those components—average, contour, and entropy—could be one of several options. Minimum and maximum (min-max) values are a straightforward choice for the contour. Similarly, the standard deviation (SD) or confidence interval (CI) could be used. These three components interact, and each choice of a triplet—average, contour, and entropy— will result in a distinct compressed classifier behavior. The goal is to discover the optimal triplet that can ignore the original data after extracting the representative contour while still being able to evaluate future series satisfactorily. We regularly apply the arithmetic average and classical entropy in our work and strive to find the best contour.

The steps in this method are further simplified below:

1. First divide the raw data into cycles.
2. Calculate the average graph line from the given N classified cycles.
3. Construct one contour for each of the K distance methods.
4. Calculate the entropy for each pair of classified cycle and contour.
5. Select the contour with the highest prediction power.

For classification, we start with a supervised learning strategy in which each time series is labeled as one of two classes. To demonstrate, the time series are year-long records of temperature samplings, labeled as positive if the corresponding year was an EN year, or otherwise negative, using the data set from the tests. We'll go over the process of creating the classifier in depth now, with a focus on finding the best contour.

We start with collecting raw data over N periods, with each record representing a single time unit. According to some classification standards, these cycles have already been labeled as positive or negative. The optimal contour will be determined using these categorized cycles.

There are four stages to the process. In the first stage, we determine the mean graph line representing the N provided cycles using a chosen average method. This is accomplished horizontally by averaging the values associated with the same time-unit across all N cycles. For instance, we calculate the average of the January 1st readings across the years. The average graph line will be generated if you do this for all time units. In the second stage, we choose multiple distance computation methods and create a contour for each method. Calculating the distance value for each distance method, such as the min-max difference, SD, and CI, accomplishes this. To get the contour around theaverage, we add and remove the distance value from the average line. This procedure will be repeated for alldistance algorithms. We've built multiple contours around the average line at this point. The goal now is to find the contour that works best for categorizing unclassified cycles. In phases three and four, this is accomplished. The prediction power for each contour is calculated in stage three, and the contour with the highest prediction power is chosen. This is accomplished by adding the number of examples in which each contour's prediction was correct and determining the average entropy of these correctly classified cycles for each contour. For incorrect forecasts, we follow the same procedure. We employ one entropy approach with an associated threshold value in stage four. Positive will be assigned to an unclassified cycle with an entropy value less than the threshold, and negative to the rest. We determine the entropy of the given categorized cycles for each contour. The result is a set of entropy values,where some are below the threshold and others are above it.

Machine Learning Technique

The process consists of seven stages. Stage 1 gathers training data from the IoT network, filters out extraneous records, and fills in blanks in records. We use discovery approaches in stage 2 to extract important measurementsand trends. The third stage is creating a rule for each measurement and pattern. In stage 4, we use a set of training data to examine the efficacy of each rule. A rule gets removed from the rules set if the number of times it has been executed falls below a certain threshold. The created set of rules is then checked for completeness and integrity in stage 5. Rules that contradict one another are eliminated, and rules that are absent are introduced. Stage 6 simulates the identical training data with the assumption that all of the defined rules will be followed. Lastly, the produced rules set is deployed in stage 7. At this point, the system is ready to accept the IoT traffic data in real time and automatically check it against the set of rules.

Getting the Rules for the Training Data

A typical sensor record contains the sensor ID, time-stamp, and one or more values per feature. The data acquired from the concrete processes engaged in the investigated domain is the primary source for extracting rules. The importance of IoT lies in making correct real-time decisions and reacting to security warnings, notifications, automation, and predictive maintenance in real time. We employ a consistent multi-layer approach of accumulating rules, starting with the simplest rules and progressing to the most difficult and multistage rules, to assure the completeness and integrity of the generated collection of rules. Simple rules are extracted at the feature level, and then rules extracted from a combination of any number of characteristics with a common relation, such as features ofsensors that share the same work-flow, are extracted. At this level, the created rules deal with fundamental data like maximum, lowest, average, standard deviation, median, and most common value. More sophisticated relationships, such as proportions between following values, sequence trends, and important patterns, necessitate reasoning skills and can be achieved using machine learning and data mining approaches. Measurements, thresholds, and patterns are used to create the decision trees. When pruning and analyzing these decision trees to discover the precise rule, they tend to expand quickly, using a lot of store and memory space, as well as a lot of time. The number of branches rises exponentially with the number of states, but the depth of the tree grows linearly with the number of variables. When the number of states per variable is minimal, decision trees come in handy. When the status of the variables is dependent on a threshold or complex computations, it gets more complicated. Labeling every edge and then tracing the tree route to comprehend the logic inherent in it is required to communicate this explanation. In the Internet of Things, complex event processing (CEP) engines are prevalent. They are capable of matching time series data patterns from many sources. They, however, have the same modeling challenges as trees and pipeline processing.

In the context of IoT, rule engines have two key drawbacks: their logic representation is not compact, and their utilization necessitates a lot of processing power and time. We shall address these flaws in two ways. 1. Reduce the number of decision trees and expand the scope of the search navigation, resulting in a search time that is appropriate and acceptable. 2. Optimize the tree navigation flow and process sharing by utilizing IoT properties and functionality.

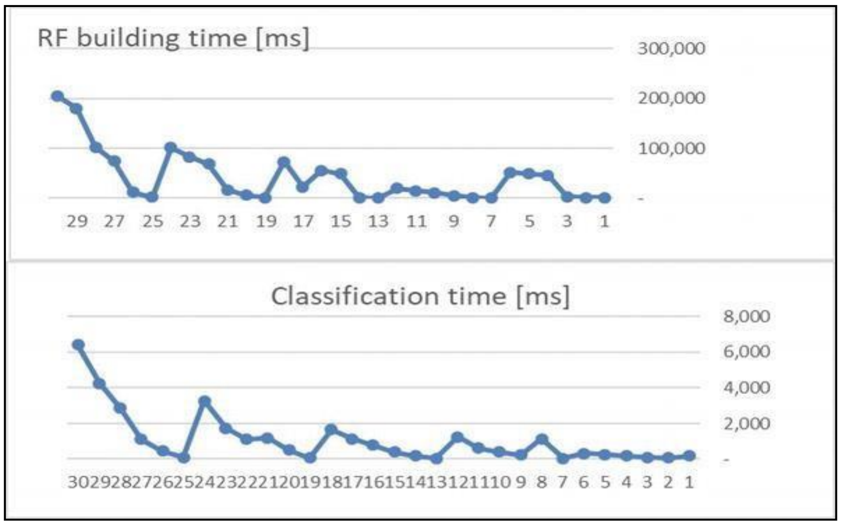
The random forest machine learning is presented in the following sections, along with many improvements that address the recognized limitations.

For training, Random Forest uses bootstrap aggregation. While a single tree's predictions are affected by noise in the training set, the average of several uncorrelated trees is not. Bootstrap sampling is a method of decorating trees by displaying alternative training sets to them. Many trees lower each tree's depth and width, saving time for pruning and analysis, which is ideal for IoT limitations. The random forest method outperforms the traditional tree decision process. However, due to the memory space and computing power required for IoT, it may still be insufficient. As a result, developing a lightweight RF method and employing IoT networking will be necessary.

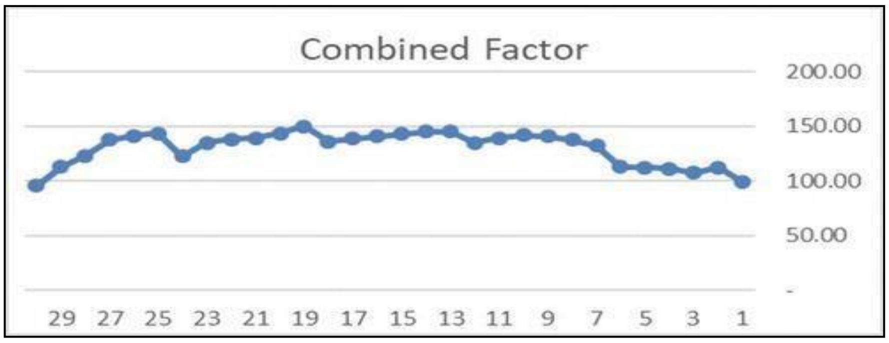
The number of K trees used to generate a random forest and the number of F randomly chosen features used to builda decision tree are the two most important factors in the technique. A big K should be utilized for huge and high- dimensional data.

The following parameters are used to estimate the performance of random forest for one core: number of trees [K], number of features [F], number of rows [R], and maximum depth [D]. The number of features has an impact on the anticipated run-time.

The Results of Running The 30 Classification Processes



The accuracy and combined results of running the 30 classification processes.



4. CONCLUSION

Endpoint clients, network devices, and cloud servers are all part of the Internet of Things (IoT) ecosystem. As a result, data transfers through the network raise a number of security risks. The recent surge in IoT applications has heightened the demand for a network architecture that can deliver rapid and secure data exchange. The use of static key authentication is a flaw in many existing networks. To make it possible for IoT devices to use automatic key update procedures and to improve security lightweight machine-to-machine (M2M) interactions is needed.

5. REFERENCES

[1] Hassan WH. Current research on internet of things (IoT) security: a survey. Comp Net. 2019; 148:283-294.
[2] Grammatikis PIR, Sarigiannidis PG, and Moscholios ID. "Securing the internet of things: challenges, threats andsolutions." Internet ofThings (2018).
[3] Harbi Y, Aliouat Z, Harous S, Bentaleb A, Refoufi A. A review of security in the internet of things. Wirel PersCommun. 2019;1-20.
[4] Adat V, Gupta BB. Security in the internet of things: issues, challenges, taxonomy, and architecture. Telecomm Syst. 2018;67(3):423-441.
[5] Dawoud A, Shahristani S, Raun C. Deep learning and software-defined networks: towards secure Iotarchitecture. Internet of Things.2018;3:82-89.
[6] Miloslavskaya N, Tolstoy A. Internet of things: information security challenges and solutions. Cluster Computing. 2019;22(1):103-119.
[7] Isha Batra, Sahil Verma, Kavita, MamounAlazab. A lightweight IoT-based security framework for inventoryautomation using wireless sensor networks.2020;33(2):e4228-10.1002/dac.4228.
[8] Menachem Domb. An Adaptive Lightweight Security Framework Suited for IoT. DOI: 10.5772/intechopen.73712.