ISBN: 978-1-6654-4502-3

Part Number: CFP21DTW-USB

2021 IEEE East-West Design & Test Symposium (EWDTS) Proceedings

Batumi, Georgia, September 10 – 13, 2021

Assessing Trustworthiness of IoT Applications Using Logic Circuits

Andrey Laputenko
National Research Tomsk State University
Tomsk, Russian Federation
laputenko.av@gmail.com

***Abstract*—The paper describes a methodology for assessing non-functional requirements, such as trust characteristics for applications running on computationally constrained devices in the Internet of Things. The methodology is demonstrated through an example of a microcontroller-based temperature monitoring system. The concepts of trust and trustworthiness for software and devices of the Internet of Things are complex characteristics for describing the correct and secure operation of such systems and include aspects of operational and information security, reliability, resilience and privacy. Machine learning models, which are increasingly often used for such tasks in recent years, are resource-consuming software implementations. The paper proposes to use a logic circuit model to implement the above algorithms as an additional module for computationally constrained devices for checking the trustworthiness of applications running on them. Such a module could be implemented as a hardware, for example, as an FPGA in order to achieve more effectiveness.**

Keywords—Internet of Things, computationally constrained devices, trust, trustworthiness, logic circuits, machine learning

I. Introduction

The widespread usage of the Internet of Things (IoT) [1] technology requires the active development of new algorithms and tools to ensure and check the correct and secure operation of devices in such networks. The use of traditional concepts of security and reliability becomes insufficient in some cases, which has led to the emerging of a more complex concept of trustworthiness, combining various requirements for software and software-hardware systems. According to the Industrial Internet Consortium, trustworthiness [2] is a degree of confidence one has that the system performs as expected with characteristics including safety, security, privacy, reliability and resilience in the face of environmental disturbances, human errors, system faults and attacks. It should be noted that this term is relatively new and periodically refined as it is discussed and studied by the scientific and technical community. Moreover, the specific requirements and restrictions on the characteristics are determined individually, depending on the area of use of the considered systems and devices of the Internet of Things.

Traditionally, for assessing and assuring key system characteristics forming a trustworthiness property in network components, the literature presents security policy approaches that establish clear solid rules for certain software use cases and scenarios. Most end-point devices in the IoT have limited computation resources, e.g. small amount of memory, not powerful CPUs, etc. Applications running on such computationally constrained devices in the IoT network require a more flexible approach. Being actively applied in practice and studied nowadays machine learning algorithms can provide such an approach. In this paper it is proposed that the level of resources consumption can affect the trustworthiness of both a computationally constrained device and an application running on such a device in the IoT.

The goal of this work is to evaluate the effectiveness of assessing trust characteristics by a logic circuit representing a machine learning model using the example of an IoT application for remote temperature monitoring. In order to this it is necessary to choose a set of parameters of resource consumption of a microcontroller with limited computing resources, reflecting the level of trust in the microcontroller application; train a machine learning model based on a dataset and synthesize a logic circuit based on it; experimentally evaluate the processing speed of input data and the classification accuracy of the synthesized logic circuit in comparison with the software-implemented machine learning model.

In this work 5 main parameters reflecting the resources consumed by the application has been chosen: heap size, stack size, CPU utilization, disk usage, power consumption. All of the above parameters can be attributed to one or several of the 5 trustworthiness characteristics that are given earlier, depending on the specific problem being solved by the application. However, sometimes certain values of parameters can be unambiguously interpreted from the point of view of trust and trustworthiness. Machine learning models having their abilities to reveal hidden patterns in arrays of data can help to treat such situations more flexible. The novelty of this paper is an approach for representing machine learning models in the form of logic circuits for the implementation in computationally constrained devices to assess non-functional requirements, for example, trust characteristics for applications running in the Internet of Things.

II. Theory

A. Machine Learning Algorithms and Models

Machine learning [3] is a set of algorithms that allow computers to very efficiently solve problems that are traditionally almost impossible to solve with classical data processing algorithms, for example, object recognition in images, e.g. faces of people. The information processing model used in this approach builds a system iteratively by "teaching" (training) it a new skill based on multiple repetitions of slightly different versions of the same process. Due to the high computing capabilities of modern CPUs, it becomes possible to process large amounts of data for more accurate training of the system and achieve the desired behavior of the system.

978-1-6654-4503-0/21/$31.00 ©2021 IEEE

There are several criteria to distinguish machine learning algorithms. The most general criterion distinguishes supervised learning and unsupervised learning. In the latter case, the model is not taught something specific, allowing it to independently select templates from the input data and generate labels for them, i.e. classifying them. This paper deals with supervised machine learning. The supervised machine learning algorithm takes stimuli and responses as input. The task is to learn how to match responses to stimuli according to the initially limited set of stimuli and reactions.

One of the most common machine learning models is an artificial neural network (ANN) [4], which copy the biological neural networks of the human brain in terms of their structure and methods of information processing. Thus, in general such networks represent a set of elements – artificial neurons, connected together by information transmission channels, called synapses. Synapses are connected to inputs and outputs of artificial neurons and information flow is directed from neurons inputs to its output. Each synapse has its own level of "conductivity" of information (weight) which allows to make some paths in the network "stronger". Usually, all neurons are divided into layers responsible for a specific function, for example, an input layer, an output layer, etc. The more layers neural network has, the more complex problems it is able to solve, but also the more resources are required to train it. A common approach is when a computationally powerful device is used to train a certain machine learning model, which is not required for the further operation of the trained model. Such a model may not contain computationally expensive operations in it. Among artificial neural networks, many varieties are also distinguished based on their structure. One of the first models was the perceptron neural network model. Nowadays, convolutional neural networks (CNN) are actively used. This model has different layers of neurons for extraction of certain properties from the input data.

The data applied to the input of a machine learning model are often called parameter values or features and are represented as a vector X of n elements x_1, x_2, ..., x_n. The expected response for a given input vector is called a label, and is denoted y, and the set of possible labels – Y. A set of parameters contains pairs <training vector, label>, where each pair is called a training example and denoted (X, y). The set of training examples is called a training set or a dataset. The main task of machine learning algorithm is to find a function $h(X)$, called a hypothesis, such that h: $X \rightarrow Y$. A common problem of machine learning algorithms is that using a training set, the model is trained with a small level of generalization, this problem is also called overfitting of model.

B. Logic Circuits as Machine Learning Models

Nowadays logic circuits are increasingly used to implement algorithms that were in the past implemented only in software, including various machine learning algorithms [5]. The advantage of using logic circuits is the ease of automatic hardware implementation of the logic circuit and the high speed of operation of such an implementation. Software simulation of the logic circuit operation is also less resource-consuming and faster [6, 7]. Logic circuit is a set of interconnected logic gates. Each logic gate has one or more inputs and one output, which can be connected to inputs of other gates, but not to any another gates output. Inputs that are not connected to any outputs are considered as primary inputs of a circuit. Outputs of any gates can be considered as primary outputs of the circuit. Logic gate implements a logic function, such as AND, OR, NAND, NOR, XOR, etc. Logic circuit which output depends only on the input values is called combinational logic circuit. In sequential circuits, an output is defined not only by input vector but by current state of the circuit which is defined by memory elements – latches or D-type flip flops. In this work only combinational circuits are considered. Logic circuit which has more than one output implements a system of Boolean functions. In order to synthesize a logic circuit one can use any logic synthesis tool, e.g. ABC [8]. The initial data for logic synthesis algorithm could be a look-up table (LUT), also called truth table, which is a table that maps input Boolean vectors to the output vectors.

The work [9] describes how logic circuits can be used as predictive models on the example of evaluating the quality of experience of the end user for a given multimedia service. The goal of this algorithm is to construct a logic circuit based on a set of parameter values and associated label values. A similar algorithm can be used to assess the characteristics of trust [6]. The algorithm includes encoding each set of parameter integer values with a Boolean vector. Then Boolean vector is associated with a Boolean vector that encodes the label value for the corresponding set of parameter values. The resulting set of Boolean vectors form a truth table (LUT) for a system of Boolean functions (SBF). In order to synthesize a logic circuit having such an SBF one can use any logic synthesis method or a logic synthesis software tool, for example, ABC [8]. The SBF constructed using initial data set contains partially specified Boolean functions, so there are some values of input variables (input vectors) for which the value of one or several functions of the SBF is undefined. For the logic synthesis algorithm, it is necessary to have a system of completely specified Boolean functions. Different ways to complete the SBF before logic synthesis step differently affect on logic circuit characteristics. Known logic synthesis algorithms during completing SBF aiming at the traditional criteria such as reducing area of a circuit and the number of a logic gates which has low correlation with classification/prediction abilities of logic circuit. Therefore, it is necessary to develop another optimization technique. In this paper it is proposed to use a pre-trained machine learning model based on the initial dataset to completely specify a system of partially specified Boolean functions.

Thus, the algorithm for constructing a logic circuit using a trained machine learning model includes the following steps:

1) Constructing a training set, namely, a sample of the values of some parameters and the corresponding label values.

2) Trainig a machine learning model M using an appropriate learning algorithm that provides the required accuracy.

3) From the entire set of input data values, a certain working data range is selected, the label values on which are obtained by simulating the trained model M.

4) The logic circuit is synthesized using one of the known methods.

It is important to note that the above technique delivers a logic circuit which has an accuracy of prediction of values the same or almost the same as a given model M for a given range of input data values. One more advantage is that described approach is applicable to any supervised machine learning model.

C. Critical Parameters of Applications for Trustworthiness Assessment

In the work [10] it is shown how the values of variables in the source code of an application can affect the trust characteristics of a given application. In this paper, it is assumed that the level of consumption of computation resources by computationally constrained devices can affect the level of trustworthiness. For an application running on computationally constrained device, one can use various parameters to estimate resource consumption, but in this paper, it is proposed to consider the following 5 most basic parameters:

- heap size (the size of the memory allocated by the application dynamically)
- the stack size (the size of the memory allocated by the application statically)
- CPU utilization (the load of the application on the central processing unit in percent)
- disk usage (the amount of memory occupied by the application for data storage)
- power consumption (the amount of power consumed by the device on which the application is running)

For some combinations of parameter values that correspond to the consumed resources by the application, it can be difficult to match the appropriate trust level for the application, even using expert assessment. For these cases, it is proposed to use pre-trained machine learning models using data obtained experimentally and / or based on expert assessment.

There are algorithms for the analysis and synthesis of logic circuits that efficiently process circuits with up to 70 inputs [8]. This should be sufficient for the synthesis of circuits processing the values of the resource consumption parameters of the computationally constrained devices, encoded with Boolean vectors, due to the limited nature of these numbers.

III. Experimental Results

The approach described in this work is demonstrated through an example of a remote temperature monitoring system via Ethernet, implemented on the LPC4088 microcontroller, which is a computationally constrained device. The characteristics of the microcontroller are shown in Table I. An application running on the microcontroller constantly reads values from the temperature sensor and constantly listens to the HTTP port for incoming connections, performing the functionality of the HTTP server. When an HTTP request appears, the application sends a message to the HTTP client with the current temperature value.

For the considered application, the reference values of the consumed parameters were determined. Then, based on these values the range of all values of the considered parameters was divided into subranges corresponding to the normal operating mode of the application and modes with deviations from the normal. These ranges are shown in Tables II and III. Table IV shows the combinations of parameter ranges and their corresponding level of application trustworthiness, according to expert assessment. For simplicity, trust labels are represented by the values 0 (trustworthy) and 1 (not trustworthy). The sign "---" in the table means that the value of the corresponding parameter does not affect the level of trustworthiness. This means that the specified set of parameters corresponds to a certain range of parameter values, at each of which the trustworthiness level values are equal. After disclosing all the intervals from the table, a set of all possible training examples was obtained, containing 108 rows, for each of which the value of the trustworthiness level is determined.

The dataset was binary encoded, and experiments with training an artificial neural network with 10 inputs and 3 layers using Keras software library were carried out on the basis of the encoded data. The script in python language for model training and running was written.

The neural network was used to build a logic circuit with 10 inputs and 1 output according to the algorithm described earlier using ABC logic synthesis tool. Then the logic circuit was translated into the C program in order to perform software simulation. The output reaction of the logic circuit when applying the set of 108 input vectors was the same as for neural network. For assessing the characteristics of data processing speed, both neural network and logic circuit were simulated 10000 times on a single input vector. The average operation time for the neural network was 0.41s, while for the software simulation of logic circuit model as program in C language – 0.03s. The experiments were carried out using laptop computer with AMD Ryzen 5 2500U CPU @ 2.0 GHz running OS Windows 10 64 bit with 12 GB RAM.

TABLE I. Characteristics of the LPC4088 microcontroller

Maximum CPU frequency	120 MHz
Flash memory size	512 KB
RAM size	96 KB
Maximum power consumption	400 mA

TABLE II. Subranges of memory consumption

	Low, KB	**High, KB**
Heap size	7,2	14,4
Stack size	40,8	81,6

TABLE III. Subranges of resources consumption

	Low	**Mid**	**High**
Power consumption	140 mA	290 mA	350 mA
CPU usage	30 KB	70 KB	90 KB
Disk usage	50 KB	80 KB	400 KB

TABLE IV. Trust levels for some combinations of parameter values

Heap size	**Stack size**	**CPU usage**	**Disk usage**	**Energy consumption**	**Trust level**
low	low	mid	mid	mid	1
high	---	---	---	---	0
---	high	---	---	---	0
---	---	high	---	---	0
---	---	---	high	---	0
---	---	---	---	high	0
---	---	---	---	low	0
low	low	low	mid	low	1
low	low	low	mid	mid	1
low	low	high	mid	high	1
---	---	---	low	---	0

An interesting question for the future work is how the complexity of logic circuit depends on the characteristics of

initial training dataset and an algorithm for machine learning model training.

IV. Conclusions

This paper describes an algorithm for synthesizing a logic circuit that reproduces the behavior of a complex machine learning model that can be integrated into a computationally constrained device to assess the level of trustworthiness for applications running on the device. A machine learning model is trained using the training dataset. The behavior of the model is further simulated on all (or part) of the possible Boolean input vectors. The resulting system of Boolean functions is used to synthesize a logic circuit. The synthesis algorithm for such a logic circuit is applicable to almost any machine learning model. The experiments carried out show that the speed of input vectors processing by the circuit exceeds the speed of the software implementation of the artificial neural network by around 10 times.

An interesting task for future work is to carry out experiments to establish correlation between various types of applications running on computationally constrained devices in the IoT network and the level of resources they consume and their effect on the level of trustworthiness of the device.

References

[1] D. Evans, "The Internet of Everything: How More Relevant and Valuable Connections Will Change the World," Cisco Internet Business Solutions Group (IBSG), Cisco Systems, Inc., San Jose, CA, USA, White Paper, 2012.

[2] Industrial Internet Consortium, "Industrial Internet Vocabulary Technical Report V 2.3," Accessed: 07.01.2021. [Online]. Available: https://www.iiconsortium.org/vocab/index.htm.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] R. D. Reed and R. J. Marks, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1998.

[5] L. Deng, "Deep learning: methods and applications, " Foundations and Trends in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014.

[6] J. López, A. Laputenko, N. Kushik, N. Yevtushenko, S. N. Torgaev, "Scalable Supervised Machine Learning Apparatus for Computationally Constrained Devices," in *Proc. 13th Int. Conf. Software Technologies* (*ICSOFT 2018*), 2018, pp. 518–528.

[7] A.V. Laputenko and S.N. Torgaev, "Comparison of the performance of different implementations of self-learning models using the example of a classification problem [Sravnenie proizvoditel'nosti razlichnyh realizacij samoobuchajushhihsja modelej na primere zadachi klassifikacii]" in *Proc. 12th Conf. Computer-aided technologies in applied mathematics*, 2018, pp. 71–72. (*in Russian*).

[8] R. Brayton, A. Mishchenko, "ABC: An Academic Industrial-Strength Verification Tool" in *Proc. Computer Aided Verification Int. Conf.* (*CAV 2010*). Edinburgh, UK, July 15-19, 2010, pp. 24–40.

[9] N. Kushik, J. Pokhrel, N. Yevtushenko, A. Cavalli, W. Mallouli, "QoE Prediction for Multimedia Services: Comparing Fuzzy and Logic Network Approaches," Int. J. Organ. Collect. Intell., vol. 4, pp. 44–64, 2014.

[10] J. López, N. Kushik, N. Yevtushenko, " Proactive trust assessment of systems as services," *in ENASE 2017*, Porto, Portugal, April 28-29, 2017, pp. 271–276.