future internet

MDPI

Article

Securing Resource-Constrained IoT Nodes: Towards Intelligent Microcontroller-Based Attack Detection in Distributed Smart Applications

Andrii Shalaginov [1,*,†] and Muhammad Ajmal Azad [2,†]

[1] Department of Technology, Kristiania University College, 0107 Oslo, Norway
[2] School of Computer Science, University of Derby, England DE22 1GB, UK; tm.azad@derby.ac.uk or majmalazad@gmail.com
* Correspondence: andrii.shalaginov@kristiania.no
† These authors contributed equally to this work.

Abstract: In recent years, the Internet of Things (IoT) devices have become an inseparable part of our lives. With the growing demand for Smart Applications, it becomes clear that IoT will bring regular automation and intelligent sensing to a new level thus improving quality of life. The core component of the IoT ecosystem is data which exists in various forms and formats. The collected data is then later used to create context awareness and make meaningful decisions. Besides an undoubtedly large number of advantages from the usage of IoT, there exist numerous challenges attributed to the security of objects that cannot be neglected for uninterrupted services. The Mirai botnet attack demonstrated that the IoT system is susceptible to different forms of cyberattacks. While advanced data analytics and Machine Learning have proved efficiency in various applications of cybersecurity, those still have not been explored enough in the literature from the applicability perspective in the domain of resource-constrained IoT. Several architectures and frameworks have been proposed for defining the ways for analyzing the data, yet mostly investigating off-chip analysis. In this contribution, we show how an Artificial Neural Network model can be trained and deployed on trivial IoT nodes for detecting intelligent similarity-based network attacks. This article proposes a concept of the resource-constrained intelligent system as a part of the IoT infrastructure to be able to harden the cybersecurity on microcontrollers. This work will serve as a stepping stone for the application of Artificial Intelligence on devices with limited computing capabilities such as end-point IoT nodes.

Keywords: cybersecurity; smart cities; smart applications; network attacks; machine learning; internet of things

Citation: Shalaginov, A.; Azad, M.A. Securing Resource-Constrained IoT Nodes: Towards Intelligent Microcontroller-Based Attack Detection in Distributed Smart Applications. *Future Internet* **2021**, *13*, 272. https://doi.org/10.3390/fi13110272

Academic Editor: Luis Javier Garcia Villalba

Received: 21 September 2021
Accepted: 26 October 2021
Published: 27 October 2021

1. Introduction

Internet of Things (IoT) is a new way of organizing domestic infrastructure and everyday automatons through portable and cheap devices which are capable of handling routine works and optimizing tasks over the large infrastructure [1]. There is a vast amount of architectures and platforms available for building ad hoc solutions ranging from single-board microcontroller units (MCU) such as Arduino and STM32, and ending with a single-board portable computer such as Raspberry Pi, Cubieboard and Orange Pi, etc. [2]. Smart Cities became reality and utilize all these technologies to make our everyday life better. Despite usefulness and incredible automation capabilities, there exist serious threats to IoT in both physical and cyber realms [3]. F-secure, a security firm estimated that cyberattacks on IoT devices are increasing at a rapid speed, i.e., more than 2.9 billion attack events that involve IoT devices were recorded in 2019 [4].

IoT end-node devices are designed to operate with a limited amount of memory, computational capabilities, and energy [5,6]. They can perform specific simple operations

helping to automate everyday tasks, while human resources can be free to execute other, more important tasks [7]. This is where the smart-alike applications became tightly integrated into our everyday life such as smartwatches, smart fridges, climate control, house alarms, etc. With the growing popularity of interconnected devices in our everyday lives, there emerges an understanding of obvious cybersecurity threats related to the omnipresent collection of different data, including personal and sensitive information [8,9]. While it is used for creating a better environment, often, the data protection is not taken into consideration [10]. It has been shown recently that IoT systems are susceptible to cyberattacks in nearly the same ways as conventional Information and Communication Technologies (ICT) systems. Cybersecurity experts keep pace in the development of attack mitigation methods on IoT gateways/hubs like Intrusion Detection Systems (IDS) and Anti-Viruses (AV). With the emergence of intelligent data systems that use previous experience and expert knowledge, one can mitigate attacks in a fast manner using fewer resources, yet the models need to be tuned [11]. Moreover, IoT end-node devices may gather information from a wide range of trivial sensors that further can be used to form awareness about states in the whole system and support automated decision making either from the incident response perspective or police investigations [12].

However, there can be seen several problems related to the characteristics of IoT end-node devices. One challenge is resource constraints that cause irregular connection intervals influenced mostly by the availability of necessary energy required for data transmission. Another challenge that is recently appearing is the examples of adversarial Machine Learning (ML) that can be considered as a major threat to intelligent cybersecurity solutions. So, there is a need to put unnecessary computations off-chip and leave only necessary ones on-chip [13]. Conventional ML methods will have to be adapted and tuned to be able to fully incorporate constraints of IoT platforms as described by Zahoor et al. [14] as well as perform necessary data fusion and feature selection from across multiple sensors. In addition to this, it can be observed a large cyber threat landscape on IoT platforms, including conventional network and malware infections. While malware for IoT end-node devices was considered to be nonsense a few years ago, the huge impact from the Mirai botnet proved that such malware has already been developed [15].

The contribution of this paper is two-fold. First, it presents a way of protecting IoT end-node devices from cyberattacks using data analytics. Recent research showed that such an approach is technically sound and can be incorporated to preserve confidentiality, integrity, and availability of the system and data. ML gives flexibility since exact network attacks rules need to be updated regularly, which is not possible in remote locations. However, the current challenge that we foresee is in the ability to deploy such models only on the level of IoT gateways that have sufficient resources for the training and testing phase. Second, we propose a framework of an intelligent system that might be used in the IoT ecosystem to support the detection of network attacks, including the active involvement of IoT end-nodes (in contrary to more powerful IoT gateways). In particular, the proof-of-concept includes the Artificial Neural Network (ANN) that is trained on the Arduino with 2 KBytes of RAM (Random Access Memory) to detect network attacks. Finally, the analysis of the current needs of intelligent IoT infrastructure protection is provided to highlight a particular way of achieving a trade-off between intelligent security systems and qualitative delivery of services. A potential application of advanced data analytics to combat the crimes in Smart Cities was presented in a winning idea during Interpol Thinkathon 2018 competition (https://www.politiforum.no/artikler/norsk-forskning-pa-framtidens-politi-til-topps-i-interpol/449053, accessed on 21 September 2021) by the NTNU team, where the idea of advanced data analytics in Smart Cities has been presented.

The paper is organized as follows. Section 2 gives an overview of the existing smart applications and modern community-accepted IoT components currently available on the market. Aspects of the utilization of Machine Learning on the Internet of Things end-node are explored in Section 3. A suggested use case scenario and methodology of the IoT-based Neural Network training is given in Section 4. Moreover, various sides of complexity and

applicability in the resources-constrained environment have been studied with respect to ANN. Corresponding training and testing phases together with computing performance implications are evaluated in Section 5. Finally, discussions of the achieved results and conclusions are given in Section 6.

2. Background Literature Review

The purpose of this section is to give an understanding of the particular applications that might be vulnerable to cyberattacks, as well as to demonstrate what are the specific components that are community accepted and can be freely used in any kind of smart applications.

2.1. Security Concerns in Smart Applications

In general, the whole range of smart applications includes any kind of hardware and software that can be used to improve living standards, facilitate everyday tasks, and simplify the interaction between humans and the surrounding environment. This can include personal health monitoring devices, smart homes, smart environments, smart factories, smart grids, etc. [16]. The main idea is to utilize multiple sensors of the information to model the surrounding context of the environment and an optimized defined model according to requirements, often adaptive.

A general IoT ecosystem normally has the following components, independently from the smart application, as presented by Mujica et al. [17]: *sensors*—simple passive devices that perform reading of measurements (temperature, brightness, etc), *actuators*—active devices that can perform actions (different servo-based elements), *IoT nodes*—end-point devices with limited computational capabilities (CPU $\approx$ MHz) that are used to collect data, send measurements and often work using batteries or solar panels, *IoT gateways*—portable devices having the functionality of low-end personal computers (CPU $\approx$ GHz), performing data processing and aggregation tasks, *Data Lakes*—redundant storage used for collecting historical data, processing and giving access to customers through the Internet. Part of the ecosystem is software that defines particular functionality and the system's response to environmental changes based on the sensors.

However, the growing popularity of smart applications made them vulnerable to multiple attacks that have been successfully demonstrated during the last years as described by Deokirikar et al. [18]. The Mirai botnet demonstrated that the security and impact of the attacks on the IoT should not be underestimated [15]. The attack was successful due to multiple factors, including weak and default passwords, insufficient or absent implementation of industrial security standards, lack of privacy-by-design. To resemble the whole picture, it was studied how well privacy is implemented in the smartwatches by famous brands [10]. Due to the lack of such an omnipresent design feature, multiple users can be affected, especially, when the smartwatches are used by kids. A similar case was known from the German doll called *"Cayla doll"*, which is susceptible to attacks, also including eavesdropping over wireless communication [19]. The discovery of the indicators of compromise led to the ban of dolls and it has been pulled out of the market.

Considering the aforementioned security concerns, there has been created a web portal *"Shodan"* (https://www.shodan.io/, accessed on 21 September 2021) to monitor devices that are connected to the Internet globally. Searching for the generic keywords *"Internet of Things"* we can get information about 9747 devices across the globe as of 10 February 2020. A corresponding search for the *"webcam"* resulted in 4598 different devices currently online. It also can be seen that in both cases, many devices use some kind of HTTP webserver to deliver the information. In a matter of fact, even MCU is capable of running web servers that can be accessed externally such as Arduino (https://www.arduino.cc/en/Tutorial/WebServer, accessed on 21 September 2021). Therefore, the amount of connected IoT devices is growing and the level of their capabilities is approaching consumer personal computers.

Cybersecurity became one of the important requirements when it comes to the design of Information and Communication Technologies (ICT) systems, including IoT. It covers

both physical aspects such as sensors and actuators in remote locations and cyber aspects, i.e., data. The data can be way more valuable than the price of the hardware and installed software [20]. Therefore, in this work, we consider data as one of the values of the smart applications and, therefore, keep in mind the following states that the data in the ICT system can be found in *data-at-rest*, *data-in-transfer*, *data-in-use.* Over the last decades, there have been developed a huge number of cybersecurity solutions that are available for personal computers as well as large-scale business solutions to mitigate basic threats, e.g., malware and network attacks. However, we have to highlight that the limited computational capabilities of the IoT devices in smart applications bring new challenges to ensuring the protection of data in the aforementioned three states.

2.2. Data-Related Capabilities of the IoT Ecosystem Components

The key to the successful identification of attacks on the IoT ecosystem is the availability of the necessary data and corresponding Indicators of Compromise. Taking into consideration a generic structure of the IoT ecosystem mentioned earlier, we focus only on *IoT nodes* and *IoT gateways*, which are more susceptible and vulnerable to cyberattacks. Those devices often have multiple communication capabilities and are located at the forefront of the anticipated attacks. On the one side, the *Data Lakes* have enough computational capacities to withstand large-scale Distributed Denial-of-Service (DDoS) attacks using enterprise solutions. On the other side, *sensors/actuators* have no functionality that can be affected-their ways to check the readings to prevent erroneous actions. Therefore, those can be only sabotaged physically, however, cannot be performed unnoticed due to multiple security measures [21].

To ensure reproducibility of the study, the range of devices that we consider are well-maintained in the community, open projects available for a wide range of customers to build end-point solutions. The main research question is to understand the capabilities of those devices related to data processing and the anticipated introduction of data analytics. The amount of data that can be generated in the IoT ecosystem has already become susceptible to the challenges affiliated with the Big Data paradigm: Velocity, Variety, Veracity, and Variability [3]. It means that one will need to develop novel cybersecurity defense mechanisms in IoT focusing on the most important traces of traffic attributed to attacks.

As mentioned before, we are looking into the capabilities of single-board MCU and single-board microcomputers or System on a Chip (SoC) from the perspective of data analytics to combat cyberattacks. Both are portable and can be easily placed on-site as well as out in a remote location with limited availability of power supply either through batteries or solar panels. Considering the popularity aspects and community-based maintenance of the projects, AVR-based Arduino can be treated as the most popular single-board micro-controller (*IoT node*) with ARM-based Raspberry Pi/Orange Pi representing single-board micro-computer (*IoT gateway*) [22]. From the cybersecurity perspective, data can be protected on the Linux-based IoT gateway using tools available for Unix such as Clam AV for malicious software detection, encryption available for Linux [23] and RPiDS [24] for IDS.

However, the application of such measures on *IoT end-nodes* is extremely limited. There is no OS, yet rather firmware that defines strict routine of initialization function SETUP() and iterative function LOOP() [25]. The only cybersecurity solution that is available and being tested for AVR is Arduino Crypto library [26]. Therefore, it is necessary to establish an understanding of what kind of data analytics for security can be run on IoT nodes and what should be moved to IoT gateway for the sake of ensuring primary services availability and data protection. Therefore, it is imperative to see the differences in the computational and data processing capabilities of both IoT nodes and IoT gateways. The comparison of the two most common, widely used, and cheap devices are given in Table 1.

From the comparison of both devices, it is clear that IoT gateway devices can participate in full-scale data processing and analytics. However, IoT node with limited computa-

tional capabilities shortens the range of additional functionality that can be implemented in addition to primary functionality.

Table 1. Comparison of popular MCU (left) and SoC (right).

Characteristics	Arduino Uno Rev3 [1]	Orange Pi One [2]
CPU frequency	16 Mhz	4-core 1 GHz
Flash memory	32 KBytes [3]	None
RAM	2 KBytes	512 MB DDR3
EEPROM	1 KByte	None
Operating Voltage	3.3–5 V	5 V
SD card extension	Possible [4]	Yes
NAS/USB HDD	No	Yes
Network	Possible [5]	Ethernet

[1]: https://store.arduino.cc/arduino-uno-rev3 (accessed on 21 September 2021); [2]: http://www.orangepi.org/orangepizero/ (accessed on 21 September 2021); [3]: of which 2 KB is used by bootloader; [4]: SD card extension shield; [5]: Ethernet card extension shield.

3. Machine Learning on the Internet of Things: State of the Art and Implications

Generally speaking, ML represents a concept of training computer systems based on the previous historical information to detect or predict patterns that have not been seen before as presented in Kononenko et al. [27]. ML has found multiple applications in security such as malware detection, network attack detection, spam detection, etc. The general ML approach is shown in Figure 1.

1. **Training**. This is the process of building the Intelligent Classifier that can help to perform similarity-based attacks classification and detection:
   - **Data Pre-processing**. The raw characteristics such as files' static and dynamic properties, network traffic packet, etc have to be harvested in a methodological reproducible manner.
   - **Feature Construction**. Extraction of the relevant and selection of the best numerical indicators that can differentiate different entry patterns. The quality of the features will define the efficiency and effectiveness of the whole model.
   - **Model Training**. During this step, the selected Machine Learning method is being trained.
2. **Testing**. This step helps to determine the particular class (e.g., malicious or benign) of a data piece that needs to be classified such as a file or network traffic packet:
   - **Pre-processing**. A set of raw characteristics is being aggregated in a way identical to *Training: Data Pre-processing* step.
   - **Feature measurement.** The raw data characteristics are extracted according to the defined previously features properties.
   - **Classification/Decision Making**. Similarity-based identification using the model constructed during the *Training: Model Training* step.

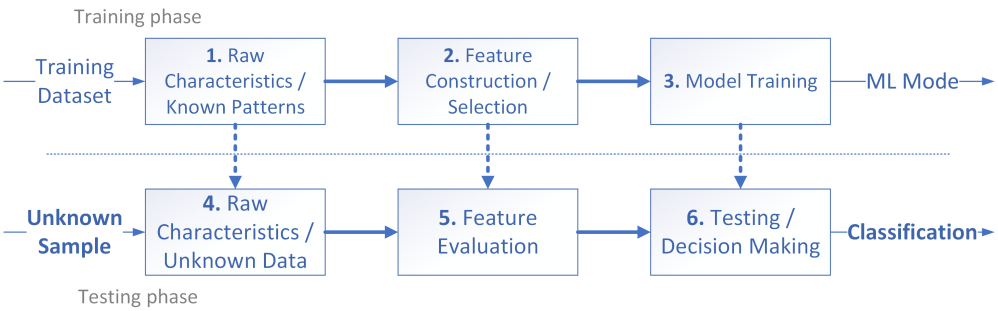

Figure 1. A general Machine Learning routine.

3.1. Community-Accepted Machine Learning Models

ML methods have been successfully used before in different tasks related to cybersecurity and data protection. A recent survey by Stam et al. [28] demonstrated how different models can be applied. The current state of the art in ML [27] includes methods like Hidden Markov Model (HMM), Support Vectors Machines (SVM), Bayesian Networks (BN), k-Nearest Neighbors (k-NN), Decision Rules (DR), Fuzzy Logic (FL), etc. With the growth in the computational capabilities of IoT devices, an opportunity came to apply more complexity and intense data analytic everywhere. It means that there can be used optimized ML models with better accuracy, model specification, capable of handling a variety of data types and decision granularity [29]. One of the recent developments in so-called Hybrid Intelligence is when stand-alone ML models are combined to mitigate the weaknesses of each and produce a better and more generalized model. Shalaginov [30] showed that Neural Networks can be combined and used with Fuzzy Logic to produce an accurate classification model as well as descriptive fuzzy logic rules set for a human-understandable explanation of the ML classification for cybersecurity.

In recent years, the Deep Learning concept has become popular [31], basically introducing higher levels of non-linearity in the models used to describe complex real-world data. Deep Neural Networks gained popularity due to the powerful ability to model problems related to text and image processing as well as in cybersecurity. In the majority of cases, *"deep"* means adding more layers to the Neural Network, such that it considers all networks with more than 3 layers [32]. A popular ML cross-platform ML tool Weka (https://www.cs.waikato.ac.nz/ml/weka/, accessed on 21 September 2021) uses a following *"rule of thumb"*: $(number\_of\_attributes + number\_of\_classes)/2$.

When it comes to the application of those models in the IoT ecosystem, there have to be considered multiple aspects and limitations for a successful application. So, what matters for the final use is (i) *space complexity* and (ii) *computational complexity* of the train ML model that can be used to predict classes of input data. We can conclude that there exist a large number of variations of ML models, however, one would need to consider the usage of feature selection and shallower models for end-point components like IoT nodes and extensive models with a higher level of abstraction at the IoT gateway level.

3.2. Human Factor in Cyberattacks Detection in Smart Cities

A need for a security expert to understand what is happening in the IoT ecosystem and if there are any zero-day attacks. One of the ways is to deploy so-called human-oriented detection, where ML models are supplemented by a description of the detection process, for example, readable linguistic rules attributed to an attack. To do so, the decision model should be deployed on IoT node, while the descriptive model should be linked and actualized on the IoT gateway/control management system.

3.3. Existing ML Implementations for IoT

To give an understanding of the existing concepts of ML application in IoT for security, a literature study has been performed. It can be seen that this perspective has drawn enough attention in the scientific community. Yavuz [33] studied an application of Deep Learning for network attack detection using a large-scale data simulation approach. The authors successfully used a data set with 64.2 million data points to train the Deep Neural Network for the detection of IoT routing attacks. Further, Canedo et al. [34] used 4000 data samples containing device ID, sensor value, and time stamp of each data transmission to train ANN for detection of abnormalities in the IoT network. Hussain et al. [35] presented an extensive overview of the literature related to the application of ML and DL for security in IoT. Multiple challenges and opportunities have been mentioned, also include the fact the IoT devices have limited capacities. Al-Garadi et al. [36] performed another extensive literature review where it was given a detailed description of possible attacks on the IoT infrastructure and network layers and now different data can be used to ensure the IoT security through training ML and DL methods in particular. Another literature review by

Andročec et al. [37] presented findings such as the most commonly used ML methods in IoT are SVM and ANN, while 62% of reviewed research works focus on network attacks, intrusion detection in particular. Restuccia et al. [38] discussed current challenges in the age of IoT and Software-Defined networks. One of the aspects is the development of secure-by-design IoT components, where authors emphasized that it is necessary to create a dynamic framework that is capable of providing application-independent protection of cybersecurity. While many relevant works have been published in recent years, the authors of those do not consider applications of the ML methods on IoT nodes, yet rather train the models on regular computers.

From a software design perspective, one would need to incorporate the current development approaches and consider existing auxiliary libraries available for MCU. On the IoT gateway side, there can be seen multiple ML libraries, supported by the fact that those have more computational capabilities. Embedded Learning Library (ELL) by Microsoft provides a wide range of functionality to support the classification of data such that audio or images [39]. Tsai et al. [40] presented demonstrated a practical demonstration of how popular TensorFlow can be used on the Raspberry Pi platform for distributed data analytics. Another example is an ML kit on Orange Pi available from ready-to-use Operating Systems (OS) software components [41]. The general pattern that we can see is that IoT gateways usually run on Linux-alike OS and any ML libraries available for PC also can be installed on IoT gateways. A single research paper that focuses on developing resource-aware ML methods for boards with 2 KBytes RAM like Arduino is written by Kumar et al. [42].

At the IoT level, we can see that there are very few works available for single-board micro-controllers such as available for Arduino [43,44]. There is also Q-behave [45], ML library for Arduino, while it is dedicated to training an Arduino to learn simple patterns from the user and not exactly an implementation of community-accepted ML models. On the other hand, there are implementations of ANN such as ArduinoANN [46] or Neurona [47]. So, there can be seen a few, mostly experimental, implementations, yet no widely-used software products.

4. Methodology: Distributed ML-Aided cyberattacks Detection on IoT Nodes

This section describes a study of the application of ML for cybersecurity on the IoT ecosystem, in particular, network attack detection. In a matter of fact, we evaluate applicability, a computational overhead that the ML model, ANN as the most popular, can bring towards the real-world IoT applications. For the evaluation phase, we have chosen to consider the most generic components, currently used in the community and widely maintained.

4.1. Use Case and Suggested Model Overview

The literature study showed that over the last few years that can be seen a large interest in applying ML for cybersecurity hardening in the IoT domain. However, most of the suggested data analytics tasks and routines have been placed out of the IoT domain onto personal computers or the cloud. The goal of this contribution is to propose a way and initiate a demonstration of how advanced data analytics can be also partially used on the IoT nodes and not just IoT gateways. Figure 2 presents the suggested integration of ML in the IoT ecosystem independently from the application. We consider the following components: Sensors/Actuators, IoT nodes, IoT gateways, and Data Lakes with large computational capabilities at the business owner's premises. In most cases, IoT nodes and IoT gateways communicate either through custom network lines or Internet connections with the main data storage. This implies, also considering possible remote locations, that an attacker can launch malicious activities against any components of this infrastructure: *Attack 1* against IoT nodes, *Attack 2* against IoT gateway, and *Attack 3* against the company itself. From the perspective of cybersecurity measures, *Attack 3* can be mitigated by using industrial solutions like AV/IDS, *Attack 2* can be mitigated by similar solutions on the

lower scale for IoT gateways, while IoT gateways do not have a standard for such measures against *Attack 3*.

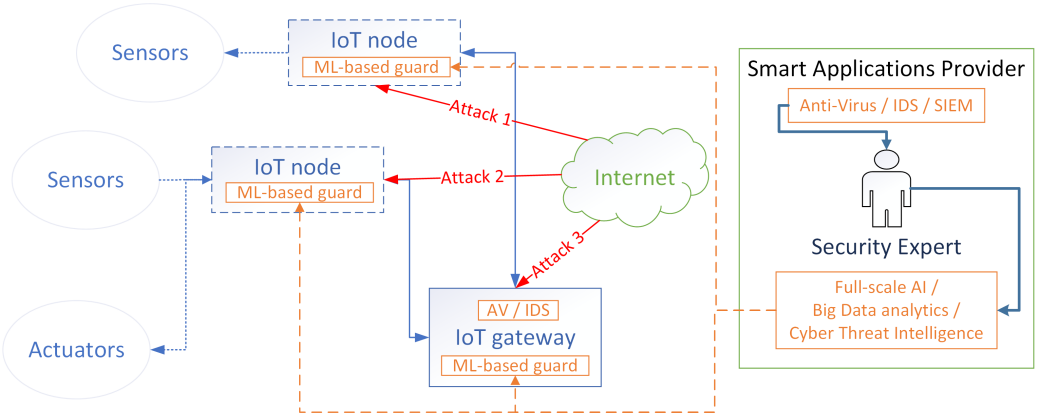

Figure 2. Machine Learning application for cybersecurity hardening in IoT ecosystem.

We propose using an advanced data analytics approach based on intelligent components across IoT infrastructure: *Big Data analytics* is done in the company's premises and application-specific ML training and inference either on *IoT gateway* or *IoT nodes*. The scenario that we consider is the application of an ML-based cybersecurity guard mechanism for detecting network attacks on the IoT infrastructure. From the data analysis perspective, KDD Cup 1999 dataset was selected as the most commonly used in the literature. It was used before in the analysis of ML complexity for network forensics by Shalaginov et al. [30]. However, it was stated that the dataset has multiple intrinsic problems with duplicate entries, so we decided to look at the improved dataset—NSL-KDD, as suggested by Tavallaee et al. [48]. Authors are aware that the datasets are considered to be obsolete from the perspective of the network attacks landscape. However, it is used for the repeatably, concept trial comparability to previous works. For the proof-of-concept demonstration, we rely on the fact that ML models have been tested on the NSL-KDD dataset and the error rates are known. Furthermore, the purpose of this article is to demonstrate the applicability of the method and not test the accuracy of ML models, which was already done before. To our knowledge, the Arduino implementation of ANN is one of the few capable of running on Arduino Uno rev3 boards. Unfortunately, it is not possible at the moment to utilize a range of other ML methods, such that are available for microcomputers from the TensorFlow Lite library [49] as it would require using different hardware components than listed in this article.

4.2. Bounding Complexity for Neural Network on IoT

Generally, data flow and computations of ML routine are easily incorporated as a singular system on personal computers or large-scale servers. From the perspective of the IoT ecosystem, it is clear that not all of the operations of ML routine can be fully deployed and utilized on micro-controllers. First, two sub-routines of the *Training* phase mentioned above are not computationally heavy. For example, parameters such as file size can be extracted using system functions and be stored in a separate features list as a numerical value. The same stands for the first two sub-routines of the *Testing* phase since those are nearly identical in both phases (feature reduction/selection is optional) [27].

However, from Figure 3, we can see that the most computationally heavy part is considered to be the third step, *Model Training*, when the hypothesis evaluation and parameters optimization happens. ANN training is performed in two steps. The *output calculation step* and *weights update step* will be as follows in case of sigmoid activation function [50]:

$$y_{actual} = sigmoid[(x_1 \cdot w_1 + \cdots + x_N \cdot w_N + b) - \theta] \quad (1)$$

$$w^i_{new} = w^i_{old} + \alpha \cdot (y_{target} - y_{actual}) \cdot x_i \quad (2)$$

Computational complexity: the first step will take N multiplications, N additions and 4 operations for activation function where N is a number of features. The second step will take approximately 4 computing operations.

Space complexity: N weights ($w_1 \cdot w_N$), a bias b, a threshold value θ and output function value y. Moreover, there has to be stored D training data samples with N features each and D class labels. In overall, it approximately requires $D \cdot (2 \cdot N + 4) \cdot M + D \cdot 4 \cdot N \cdot M$ computing operations, where M is a number of training epochs.

Once the model is trained from the input data, the internal structure of the ML model is fixed as a set of weights/parameters. During the *Testing* phase, $2 \cdot N + 4$ parameters estimation is used against previously defined features to determine a class of the input data sample. From the storage complexity perspective, it will require $N + 2$ memory units for the parameters of the ANN and N parameters of a single unclassified data sample.

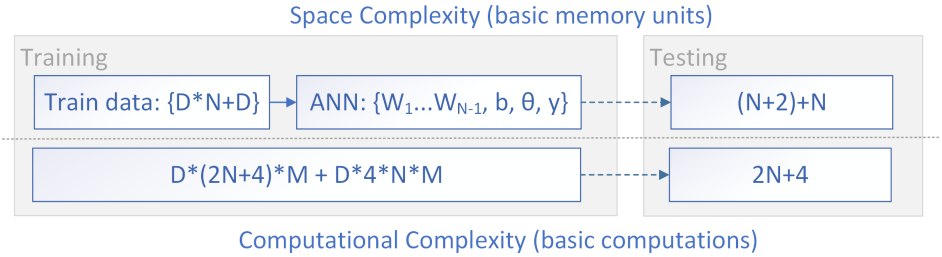

Figure 3. Example of space and computational complexity of single-layer ANN.

In the real world, we can see that training ANN might be a way to heavy task for the IoT ecosystem components. Considering the aforementioned justification, training of a single-layer ANN might have polynomial complexity of $O(N^3)$, while training will be nearly linear $O(N)$. For example, using KDD Cup 1999 full data set, it requires nearly 1110 seconds to train a Neuro-Fuzzy model on a powerful server station, while the testing phase requires 95×10^{-6} seconds per new data sample to predict the actual class as was demonstrated in Shalaginov et al. [30]. Therefore, it is important to see that the heaviest parts of the computations in ML routine, i.e., *Training*, should be placed off IoT nodes to IoT gateways, while the prediction, i.e., *Testing*, can still be left at the IoT node level.

5. Experimental Design Analysis of Results

This section presents an overview and implications of the training and testing phases of the ML approach to be used on the IoT node.

5.1. Training–Building a Model

Features in the NSL-KDD dataset represent following characteristics: (i) basic TCP connection, (ii) packet content features, (iii) time window-based traffic features. The aforementioned study [30] used the following features for the experiments (id - feature name): *6-dst_bytes, 41-dst_host_srv_rerror_rate, 10-hot, 12-logged_in, 14-root_shell, 22-is_guest_login, 29-same_srv_rate, 37-dst_host_srv_diff_host_rate, 5-src_bytes*. Two types of the most common traffic found in the dataset were selected: *"normal"*—Class ID 0 and *"neptune"* (sub-type of DOS)—Class ID 1. For *software part*, *ArduinoANN* implementation [46] (on Arduino IDE v1.8.9) of neural network was used with the following parameters: training epochs-max 2^{32-1}, learning rate = 0.3, momentum = 0.9, input nodes = 9, hidden nodes = 5, weights initialization = 0.5, error threshold = 0.001. The original implementation was designed to recognize binary data from digits on LCD array to number, while it was extended to be able to handle real-valued data from KDD dataset. For *hardware part*, Arduino Uno Rev3 board and USBtinyISP programmer with connection at 9600 baud rate for serial monitor was utilized. The *space complexity* evaluation is given in Table 2 and basically represents that the training data are stored as 20 data samples with 9 features in 4-Byte real value and a class label in 2-Byte integer. In reality, the difference between storing 2 samples and

20 samples will be equal to 720 Bytes/18 = 40 Bytes per each network packet stored in the memory for the model training. The main limitation at the training phase is the limited amount of SRAM (2 KBytes).

Table 2. Space complexity of the ANN model training on Arduino Uno Rev3.

Space Compexity	Flash, Bytes (%)	SRAM, Bytes (%)
Full memory	32,256 (100%)	2048 (100%)
2 training packets (9 f.)	8276 (25%)	970 (47%)
20 training packets (9 f.)	8960 (27%)	1690 (82%)

Table 3 demonstrates the performance evaluation and time complexity of different epochs. It took 1 second and 326 milliseconds to train the ANN model with 16 epochs (until the error threshold has been reached). The *micros()* function was used to measure the time. To thoroughly evaluate the performance, we utilized the training error calculation function during each training epoch in the Neural Network: $E = \frac{1}{2 \cdot M} \cdot \sum_{i=1}^{M} (y^i_{target} - y^i_{actual})^2$ with error terminating criteria: $\epsilon = 10^{-3}$. This is similar to regression-based loss function Mean Square Error (MSE) used in popular Keras library [51] for any type of classification problems. Such an approach gives a better understanding of per-epoch performance improvement in comparison to conventional classification-based metrics such that accuracy where one would need to utilize *round()* function to approximately determine where the tested sample belongs to Class 0 or Class 1. This is a proof-of-concept work and we believe that setting higher values of ϵ will have a negative impact on the stopping criteria in a real-world case with erroneous and imprecise data. Furthermore, this helps to take into consideration the trade-off between accuracy and training time as some of the real-world problems cannot guarantee 100% accuracy while using the Machine Learning community-accepted ANN method. In this work, we did not focus on the performance aspects of ANN as this was heavily tested and validated before. However, we can state that the main parameters that generally influence the performance and training time are number of training epochs M, training error threshold ϵ and number of layers in the neural network. As of now, ANN implementation for Arduino UNO Rev3 cannot incorporate higher complexity due to the resource-constrained environment issues with the memory considerations stated in Table 2. Having a bigger size of memory would allow utilization of more advanced ML techniques.

Table 3. Computation complexity and performance of the ANN training phase.

Epoch ID	1	10	16
Output Error	2.08758	0.01587	0.00946
Time (per epoch), 10^{-6} s	28,752	29,808	29,756

So, it can be seen that ANN can be trained on Arduino UNO Rev3 with limited training data in a reasonable amount of time. To increase the number of training samples, the possible solution is to utilize the SD card and discard the training data once the model is trained, yet this will bring additional overhead.

5.2. Testing–Attack Detection Phase

While there can be found many ML libraries for IoT gateways, to authors' knowledge, there have not been tested and developed many ML approaches for end-point IoT nodes that can contribute towards this. Most of the works mentioned earlier use data analytic and ML for the analysis of data extracted from IoT devices, which is done on more powerful machines. The testing phase includes a comparison of the network packet that arrives against the previously training model. It is similar to IDS/Intrusion Prevention Systems

(IPS) systems when the network traffic is inspected and being compared against predefined signatures. Table 4 demonstrates the difference between model accuracy and time to classify a single network packet, while measured during the first/last epoch.

Table 4. Example of predicted ANN output ("Output"), normal/attack class ("Target") and used time ("Time") for the classification of first 10 network packets.

ID	1st Epoch			16th Epoch		
	Target	**Output**	**Time, µs**	**Target**	**Output**	**Time, µs**
1	0	0.39667	1196	0	0.03863	1388
2	0	0.39667	1896	0	0.03863	1932
3	1	0.70681	1752	1	0.97040	1836
4	0	0.27704	1176	0	0.02999	1176
5	0	0.32847	1668	0	0.02803	1668
6	1	0.75069	1792	1	0.98339	1856
7	1	0.70681	1752	1	0.97040	1836
8	1	0.70512	1756	1	0.96993	1828
9	1	0.70606	1752	1	0.97020	1820
10	1	0.70662	1740	1	0.97035	1836

To practically reflect the *computational complexity* of the training phase, we can say that the average time needed to evaluate a single network packet based on 9 characteristics on the Arduino UNO Rev3 using trained ANN model is 1643 µs. It will be equal approximately to 600 network packets/s. However, these are the ideal conditions and more study is needed using an actual network extension board. From the study [30], we can see that it takes up to 40 µs for an ML model to classify a questioned network traffic packet on a multi-core system. At the same time, the extensive benchmark of the different extension shields for Arduino UNO Rev3 gives the data transfer rate ranging from 79.27 KBytes/s up to 329.60 KBytes/s. Using a rough estimation, we can see that the following packet rate range can be expected [30]:

$$Rate_{min} = \frac{79.27 \times 10^3 \text{ Bytes/s}}{84 \text{ bytes/packet}} \approx 943 \text{ packets/s} \quad (3)$$

$$Rate_{max} = \frac{329.60 \times 10^3 \text{ Bytes/s}}{1538 \text{ bytes/packet}} \approx 214 \text{ packets/s} \quad (4)$$

At the same time, the *space complexity* for the ready-to-use ANN model will be as follows: $(N+1) \cdot H$ weights in the hidden layer, $H+1$ weights in the output later, number of input nodes N, number of hidden H and number of output layer O, where N is a number of input features, H is a number of hidden layers, O is a number of output nodes. For this specific case, we can evaluate the model will require, respectively, 50 4-Byte real weights, 6 4-Byte real weights, 1 2-Byte integer, 1 2-Byte integer, and 1 2-Byte integer. Such guesstimate gives roughly 230 Bytes needed to store the ANN model on the IoT node (Arduino UNO Rev3).

Digital Evidences Analysis in the IoT ecosystem. One of the important considerations while using Computational Intelligence in Cybersecurity is the ability to derive an explanation of the cyber incident. Moreover, the data can be used as evidence in case of an investigation. From the perspective of IoT, this is a challenging task due to: (i) limited data storage, (ii) limited logging (username, timestamps, non-repudiation), (iii) mostly proprietary protocols and data formats, (iv) absence of wide support by Digital Forensics tools. Considering all this, it is also important to have intelligent models to possess some level of explainability of the decision being made, for example, through linguistic and human-understandable approaches like Fuzzy Logic rules in addition to numerical ANN weights on IoT nodes.

6. Conclusions & Discussions

Internet of Things devices became a vital part of our everyday activities bringing flexibility, convenience, and smart application in many domains. While this is undoubtedly disruptive technology with great potential for live standards improvement, there exist many security challenges that also can be found in ICT components. However, it has been shown that IoT devices are more susceptible to attacks due to reduced capabilities for implementing security standards, as well as limited functionality that can be deployed to protect the infrastructure. This paper studies the current state of the art related to IoT end-node devices security measures, the applicability of Machine Learning, and data analytic existing standards. We propose a concept of the intelligent model training for IoT end-node device-based network attacks detection, in addition to commonly used IoT gateways cybersecurity measures. Moreover, a proof-of-concept is implemented on MCU Arduino Uno Rev3 with 2KByte RAM to demonstrate how the Artificial Neural Networks can be used for intelligent network attack detection with a low error rate. It is clear that based on the selected features, the model can be trained even on IoT end-node in nearly 1 s using 16 training epochs to achieve necessary accuracy, while it takes around 2 ms to classify previously unseen network packets. However, the training phase should be placed off-chip on a more powerful IoT gateway device, while the training can be done on IoT nodes using weights of the already trained model. This ongoing work is intended to serve as a stepping stone for future research in the practical applications of ML on IoT end-node device nodes.

Author Contributions: Conceptualization, A.S. and M.A.A.; methodology, A.S. and M.A.A.; software, A.S.; validation, A.S.; formal analysis, A.S. and M.A.A.; investigation, A.S.; resources, A.S.; data curation, A.S.; writing—original draft preparation, A.S. and M.A.A.; writing—review and editing, A.S. and M.A.A.; visualization, A.S.; supervision, A.S. and M.A.A; project administration, A.S.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: The authors would like to acknowledge support and funding provided by the Department of Technology, The School of Economics, Innovation, and Technology at the Kristiania University College.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Scott, A. 8 Ways the Internet of Things Will Change the Way We Live and Work. Available online: https://www.businessnewsdaily.com/4858-internet-of-things-will-change-work.html (accessed on 21 September 2021)
2. Miller, W. Comparing Prototype Platforms: Arduino, Raspberry Pi, BeagleBone, and LaunchPad. Available online: https://www.electronicproducts.com/comparing-prototype-platforms-arduino-raspberry-pi-beaglebone-and-launchpad/ (accessed on 21 June 2019).
3. Shalaginov, A.; Kotsiuba, I.; Iqbal, A. Cybercrime Investigations in the Era of Smart Applications: Way Forward Through Big Data. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 4309–4314.
4. Cyberattacks on IOT Devices Surge 300% in 2019, 'Measured in Billions', Report Claims. 2019. Available online: https://www.oodaloop.com/briefs/2019/09/16/cyberattacks-on-iot-devices-surge-300-in-2019-measured-in-billions-report-claims/ (accessed on 10 February 2020).
5. Arshad, J.; Azad, M.A.; Abdellatif, M.M.; Rehman, M.H.U.; Salah, K. COLIDE: A collaborative intrusion detection framework for Internet of Things. *IET Netw.* **2019**, *8*, 3–14. [CrossRef]
6. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time Intrusion Detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
7. Ranger, S. What Is the IoT? Everything You Need to Know about the Internet of Things Right Now. Available online: https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/ (accessed on 21 September 2021)
8. Azad, M.A.; Bag, S.; Parkinson, S.; Hao, F. TrustVote: Privacy-Preserving Node Ranking in Vehicular Networks. *IEEE Internet Things J.* **2019**, *6*, 5878–5891. [CrossRef]

9. Azad, M.A.; Bag, S.; Hao, F.; Salah, K. M2M-REP: Reputation system for machines in the internet of things. *Comput. Secur.* **2018**, *79*, 1–16. [CrossRef]
10. Council, N.C. WatchOut—Analysis of Smartwatches for Children. Available online: https://www.conpolicy.de/en/news-detail/watchout-analysis-of-smartwatches-for-children/ (accessed on 24 June 2019).
11. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 29 May–1 June 2018; pp. 371–390.
12. Yamin, M.M.; Shalaginov, A.; Katt, B. Smart Policing for a Smart World Opportunities, Challenges and Way Forward. In Proceedings of the Future of Information and Communication Conference, San Francisco, CA, USA, 5–6 March 2020; pp. 532–549.
13. Shalaginov, A.; Semeniuta, O.; Alazab, M. MEML: Resource-aware MQTT-based Machine Learning for Network Attacks Detection on IoT Edge Devices. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion, Auckland, New Zealand, 2–5 December 2019; pp. 123–128.
14. Zahoor, S.; Mir, R.N. Resource management in pervasive Internet of Things: A survey. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 921–935. [CrossRef]
15. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the mirai botnet. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
16. Analytics, I. The 10 Most Popular Internet of Things Applications Right Now. 2016. Available online: https://bigdatanomics.org/index.php/iot-cloud/235-the-10-most-popular-internet-of-things-applications (accessed on 22 June 2019).
17. Mujica, G.; Portilla, J. Distributed Reprogramming on the Edge: A New Collaborative Code Dissemination Strategy for IoT. *Electronics* **2019**, *8*, 267. [CrossRef]
18. Deogirikar, J.; Vidhate, A. Security attacks in IoT: A survey. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, Tamil Nadu, India, 10–11 February 2017; pp. 32–37.
19. Torano, C. *IoT for Kids: Cayla Doll Exploit*; Technical Report; Abertay University: Dundee, UK, 2014.
20. Saif, I. Striking a Balance between Extracting Value and Exposing Your Data. 2013. Available online: https://www.ft.com/content/35993dce-933a-11e2-9593-00144feabdc0 (accessed on 25 June 2019).
21. Barcena, M.B.; Wueest, C. Insecurity in the Internet of Things. Available online: https://docs.broadcom.com/doc/insecurity-in-the-internet-of-things-en (accessed on 25 June 2019).
22. Difference between Raspberry Pi vs. Orange Pi. Available online: https://www.geeksforgeeks.org/difference-between-raspberry-pi-and-orange-pi/ (accessed on 25 June 2019).
23. Top 5 Raspberry Pi Network Security Tips for Beginners. Available online: https://www.raspberrypistarterkits.com/guide/top-raspberry-pi-network-security-tips-beginners/ (accessed on 10 June 2019).
24. Sforzin, A.; Mármol, F.G.; Conti, M.; Bohli, J.M. RPiDS: Raspberry Pi IDS—A Fruitful Intrusion Detection System for IoT. In Proceedings of the 2016 International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18–21 July 2016; pp. 440–448.
25. Arduino Language Reference. Available online: https://arduinogetstarted.com/arduino-language-reference (accessed on 25 June 2019).
26. Arduino Cryptography Library. Available online: http://rweather.github.io/arduinolibs/crypto.html (accessed on 24 June 2019).
27. Kononenko, I.; Kukar, M. *Machine Learning and Data Mining*; Horwood Publishing: Cambridge, UK, 2007.
28. Stamp, M. A Survey of Machine Learning Algorithms and Their Application in Information Security. In *Guide to Vulnerability Analysis for Computer Networks and Systems*; Springer: Berlin, Germany, 2018; pp. 33–55.
29. IoT Hardware Guide. 2019. Available online: https://www.postscapes.com/internet-of-things-hardware/ (accessed on 5 June 2019).
30. Shalaginov, A.; Franke, K. Big data analytics by automated generation of fuzzy rules for Network Forensics Readiness. *Appl. Soft Comput.* **2017**, *52*, 359–375. [CrossRef]
31. Berman, D.S.; Buczak, A.L.; Chavis, J.S.; Corbett, C.L. A survey of deep learning methods for cyber security. *Information* **2019**, *10*, 122. [CrossRef]
32. A Beginner's Guide to Neural Networks and Deep Learning. Available online: https://wiki.pathmind.com/neural-network (accessed on 3 June 2019).
33. Yavuz, F.Y. Deep Learning in Cyber Security for Internet of Things. Ph.D. Thesis, Istanbul Sehir University, Istanbul, Turkey, 2018.
34. Canedo, J.; Skjellum, A. Using machine learning to secure IoT systems. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, North Island, New Zealand, 12–14 December 2016; pp. 219–222.
35. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *arXiv* **2019**, arXiv:1904.05735.
36. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.; Du, X.; Guizani, M. A survey of machine and deep learning methods for internet of things (IoT) security. *arXiv* **2018**, arXiv:1807.11023.
37. Andročec, D.; Vrček, N. Machine Learning for the Internet of Things Security: A Systematic. In Proceedings of the 13th International Conference on Software Technologies, Porto, Portugal, 26–28 July 2018; p. 97060. [CrossRef]

38. Restuccia, F.; D'Oro, S.; Melodia, T. Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet Things J.* **2018**, *5*, 4829–4842. [CrossRef]
39. Embedded Learning Library (ELL). 2018. Available online: https://microsoft.github.io/ELL/ (accessed on 17 June 2019).
40. Tsai, P.H.; Hong, H.J.; Cheng, A.C.; Hsu, C.H. Distributed analytics in fog computing platforms using tensorflow and kubernetes. In Proceedings of the 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, Korea, 27–29 September 2017; pp. 145–150.
41. Build Machine Learning Environment on OrangePi Zero Plus (arm64). 2018. Available online: https://github.com/hankso/OrangePi-BuildML (accessed on 10 June 2019).
42. Kumar, A.; Goyal, S.; Varma, M. Resource-efficient machine learning in 2 KB RAM for the internet of things. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1935–1944.
43. Protecting the Three States of Data. 2016. Available online: https://www.sealpath.com/blog/protecting-the-three-states-of-data/ (accessed on 26 June 2019).
44. Mellis, D.A. ESP (Example-Based Sensor Predictions). 2017. Available online: https://github.com/damellis/ESP (accessed on 21 June 2019).
45. Śmigielski, M. Machine Learning Library for Arduino. 2014. Available online: https://github.com/smigielski/q-behave (accessed on 25 June 2019).
46. A Neural Network for Arduino. 2012. Available online: https://www.bilibili.com/read/cv3119927 (accessed on 14 June 2019).
47. Moretti, C.B. Neurona—Artificial Neural Networks for Arduino. 2016. Available online: https://github.com/moretticb/Neurona (accessed on 23 June 2019).
48. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
49. TensorFlow. Deploy Machine Learning Models on Mobile and IoT Devices. Available online: https://www.tensorflow.org/lite (accessed on 15 October 2021).
50. Flood, I.; Kartam, N. Neural networks in civil engineering. I: Principles and understanding. *J. Comput. Civ. Eng.* **1994**, *8*, 131–148. [CrossRef]
51. Keras-Losses. Available online: https://keras.io/api/losses/ (accessed on 15 October 2021).