IoT Device Identification and Cybersecurity: Advancements, Challenges, and an LSTM-MLP Solution

Shaya A. Alshaya
Computer Science Department, College of Sciences and Humanities at Al-Ghat, Majmaah University, Saudi Arabia
shaya@mu.edu.sa (corresponding author)

Received: 21 August 2023 | Revised: 13 September 2023 | Accepted: 19 September 2023

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: https://doi.org/10.48084/etasr.6295

ABSTRACT

Over the past few years, there has been an undeniable surge in the deployment of IoT devices. However, this rapid growth has brought new challenges in cybersecurity, as unauthorized device deployment, malicious code modification, malware deployment, and vulnerability exploitation have emerged as significant issues. As a result, there is a growing need for device identification mechanisms based on behavior monitoring. To address these challenges, Machine Learning (ML) and Deep Learning (DL) techniques have been increasingly employed due to advances in the field and improved processing capabilities. However, cyber attackers have developed adversarial attacks that focus on modifying contexts and evading ML evaluations applied to IoT device identification solutions. This article highlights the importance of addressing cybersecurity challenges in the IoT landscape and proposes a hardware behavior-based individual device identification approach using an LSTM-MLP architecture. The proposed architecture was compared to the most common ML/DL classification techniques using data collected from 45 Raspberry Pi devices running identical software and showing promising results in improving device identification. The proposed LSTM-MLP method outperformed previous solutions, achieving an average increase in F1-Score of +0.97 and a minimum TPR of 0.97 for all devices.

Keywords-IoT devices; cybersecurity challenges; device identification; LSTM-MLP architecture; adversarial attacks

I. INTRODUCTION

The latest revolution in communication technologies and processing has paved the way for a significant increase in the deployment of Internet-of-Things (IoT) devices [1]. The versatility of these devices has led to their widespread use in various applications, including Industry 4.0, Smart Cities, homes, and healthcare, resulting in a diverse range of IoT device types to meet different scenarios. System-on-Chip (SoC) devices, such as Raspberry Pi (RPi), have become popular due to their relatively high processing power, flexibility, and cost-effectiveness, making them even more appealing than less powerful alternatives [2]. However, this increase in processing power has also brought about cybersecurity challenges, as more powerful IoT devices can be used for more powerful attacks, such as Distributed Denial of Service (DDoS) attacks or crypto jacking attempts [3]. Securing the IoT landscape, especially with the use of SoCs, is crucial to ensure its proper functioning and protection against potential threats.

A vital aspect of IoT security lies in accurately identifying each deployed device to prevent the presence of unauthorized devices. While traditional static identifiers were used, attackers can easily manipulate or duplicate them. To address this issue, many methods deal with the behavior of the device during the identification process [4]. IoT device identification based on behavior can be approached from different levels related to specific environment requirements [4]. Behavioral identification is treated by two main approaches: type identification or device model, which categorizes devices based on characteristics like network activities and running processes, and individual device identification, which distinguishes devices of the same model through low-level component analysis or radio frequency fingerprinting. Individual device identification offers the highest level of security but requires meticulous monitoring of chip manufacturing variations to differentiate between devices with identical hardware and software [5]. The analysis of hardware performance is a widely used technique, in which the behavior of components such as the CPU, GPU, or RAM is monitored during specific tasks [6]. However, attackers can exploit the usage patterns or context of the device to manipulate the values that generate the device fingerprint, thus disrupting the identification process.

In recent years, the application of Machine Learning (ML) techniques has become increasingly prominent in IoT security and is now widely used for device identification [7-8].

However, with the adoption of ML/DL techniques, adversarial attacks against these models have emerged that aim to interfere with the training process or deceive the model predictions, posing various threats to the ML/DL pipeline [9]. Attackers can poison training data, exploit vulnerabilities in testing data, or infringe privacy by inferring data from the model and its gradients [10-11]. Even ML/DL-based IoT identification solutions have not been immune to these adversarial attacks [12]. Context modifications and the use of crafted adversarial samples during the identification process have demonstrated that these solutions are also vulnerable [13-15].

This study tackled the complex challenges associated with merging hardware-based individual device identification, ML/DL techniques, and adversarial attacks, encompassing several significant contributions:

- Comprehensive threat model: A comprehensive threat model was defined that covers the entire data lifecycle, from the creation of device fingerprints to their evaluation. This model mainly involves the utilization of ML and DL techniques, considering the potential vulnerabilities and threats at each stage of this process.
- Innovative neural network architecture: This study applied a neural network architecture called LSTM-MLP (Long Short-Term Memory - Multi-Layer Perceptron). This architecture was designed to identify individual devices based on their hardware performance behavior. Instead of treating performance measurements as isolated data points, this model treats them as time series data, allowing more effective data processing and pattern extraction to improve the accuracy of device identification.
- Performance evaluation: A performance comparison was conducted to assess the effectiveness of the proposed LSTM-MLP architecture with various ML and DL classification approaches. The LwHBench dataset [16] was used, which consists of hardware performance and behavior data collected from 45 RPi devices running identical software configurations. The proposed model achieved remarkable results, with an average F1-Score of 97.5% and a True Positive Rate (TPR) of 97.2%. These results demonstrate the superiority of the proposed approach in accurately identifying individual devices based on their hardware performance behavior.

II. RELATED WORKS

A. Hardware-centric Individual Device Recognition

In [6], the variance in the GPU and CPU cycle counters within RPi devices was juxtaposed to achieve a distinctive identification of 25 devices, using XGBoost and achieving a TPR of 91.92%. In [15], GPU performance patterns were coupled with ML and DL classification algorithms, achieving accuracy levels ranging from 95.8% to 32.7% for nine batches of similar devices. In [5], variations in code execution proficiency were evaluated to recognize more than 260 indistinguishable computers. The Real-Time Clock (RTC), equipped with its unique physical oscillator, was used to identify subtle deviations in the execution capabilities of individual CPUs. This process involved comparing the CPU time counter, RTC chip, and the Digital Signal Processor (DSP) within the sound card to precisely identify identical computers. In [17], Physical Unclonable Functions (PUFs) were investigated for IoT device identification, although this study exclusively focused on hardware behavior fingerprinting driven by device performance and did not consider PUFs. Hardware-centric approaches suffer from many limitations:

- Hardware-centric focus: While this approach can be effective for certain scenarios, it may not be suitable for cases where device hardware is not readily accessible or where software-based emulation can deceive the recognition system.
- Dependency on specific hardware elements: These methods heavily rely on specific hardware elements, such as GPU and CPU cycle counters, RTC chips, and DSPs, to distinguish devices. This reliance limits its applicability to devices that possess these particular components and may not work on devices lacking them.
- Limited scalability: Some studies achieved recognition of a relatively small number of devices. This limited scalability may restrict its utility for large-scale deployments or networks with numerous identical devices.
- Varying accuracy: Using this approach, the accuracy of device identification varies widely, ranging from high to lower accuracy levels. Such variability in accuracy can be problematic for applications that require consistent and reliable identification.
- Exclusion of PUFs: These approaches did not use PUFs for IoT device identification, which is a distinct and widely investigated method in the field. PUFs offer unique advantages, including enhanced security, and their exclusion may limit effectiveness in certain contexts.

B. Context-focused Attacks

In hardware-based identification solutions, the context in which the identification code or tasks are executed can significantly influence the collected data and, consequently, the results. For example, temperature variations may affect the frequency of crystal oscillators, and hardware load can introduce delays due to process scheduling. In the hands of a malicious attacker, these context conditions can be manipulated to disrupt the identification process, rendering it unusable or generating measurements that mimic another device.

The studies mentioned in the previous section briefly addressed context-related issues that could affect the identification process. In [6], it was found that device rebooting and concurrent processes had an impact on identification results if not implementing proper process isolation mechanisms for data collection. However, the usual temperature changes based on the device load did not appear to significantly affect the results. In [15], it was shown that environmental temperatures between 26.4 ºC and 37 ºC did not have a significant impact on the identification results, but device rebooting had a considerable negative effect, reducing the accuracy to 50.3%. This study suggested that voltage variations should be considered as a future evaluation line. On the other hand, in [5], the identification application was evaluated under different CPU loads and temperatures, obtaining positive results in both scenarios, while the temperature impact analysis was only mentioned as part of future work, and no context-based experiments were performed. Despite the existing research on device fingerprinting and identification based on hardware performance behavior, none of the previous studies extensively explored the potential impact of context-focused attacks on their results. This aspect remains largely unexplored and presents a crucial area for further investigation.

C. ML/DL-focused Adversarial Attacks

Adversarial ML/DL is a research area dedicated to developing accurate and highly robust models that can withstand tampering attempts [10]. It involves the study of various attacks against ML/DL models and the defense techniques employed to enhance their security. These attacks can be categorized into different types:

- Poisoning attacks during the training process, where malicious samples are used to compromise the integrity of the model.
- Evasion attacks target the model evaluation process, attempting to deceive a legitimately trained model into misclassifying samples.
- Model extraction attacks involve an attacker trying to infer the model based on its predictions.

This study focused on evasion attacks, specifically to deceive a model trained for device identification into misclassifying a malicious device as a legitimate one. Within evasion attacks, there are two main types: non-targeted attacks, which aim to misclassify samples into any different class than their original one, and targeted attacks, which aim to make the model classify a malicious sample as a specific target class.

Numerous evasion attacks are commonly found. One of the first DL-based attacks is the Fast Gradient Sign Method (FGSM) [18], which performs one-step updates on an adversarial sample, following the direction of gradient loss in an attempt to move the sample to the boundary of a different class. The Basic Iterative Method (BIM) [19] builds on FGSM by incorporating iterative optimization. This involves applying FGSM multiple times with small perturbation steps. The Momentum Iterative Method (MIM) [20] introduces momentum in iterative FGSM or BIM to mitigate the influence of local minimum or overfitting in generating adversarial samples. Projected Gradient Descent (PGD) [21] is an extension of BIM that does not impose constraints on the iteration steps. The DeepFool L2 attack [22] minimizes the Euclidean distance between the original and adversarial samples by estimating the model decision boundary using a linear classifier. Jacobian-based Saliency Map Attack (JSMA) [23] is a common attack based on the Jacobian matrix [24] of the model to determine the sensitivity direction and perform feature selection, minimizing the number of modified characteristics from the original data sample. Boundary Attack [25] generates a random adversarial sample and optimizes the L2 norm of the perturbation to make the sample similar to the original legitimate vector while maintaining the misclassification result. Carlini and Wagner (C&W) Attack [26] proposes an optimization-based method to generate adversarial samples, which can be applied to three distance metrics: L0, L2, and L1. L0 measures the number of modified features, L2 measures the Euclidean distance between benign and adversarial samples, and L1 measures the maximum change in any feature. The Generative Adversarial Network (GAN) attack uses GAN models to generate realistic adversarial samples that can deceive the classifier.

Several defense mechanisms have been developed against such attacks [27]. These countermeasures aim to make models resilient to adversarial samples and can be classified into detection and robustness methods. Detection methods focus on identifying crafted malicious samples before evaluation, while robustness methods aim to make the model resistant to the evaluation of adversarial samples. Additionally, defense mechanisms can be attack-specific or attack-agnostic, depending on whether they target improving resilience against a specific attack. Adversarial Training [28] is a widely used defense technique that involves training models with malicious samples to reduce the impact of attacks that generate them. Knowledge distillation [29] is another method applied to improve robustness during training, by generating smaller models using the base model outputs as features, leading to smoother decision boundaries and reduced sensitivity to adversarial samples [30].

In [31], the effects of different non-targeted and targeted adversarial attacks (such as FGSM, BIM, PGD, and MIM) were investigated on a CNN used for radiofrequency-based individual device identification. Similarly, in [32], the resilience of network-based IoT identification ML models was assessed against adversarial samples generated using FGSM, BIM, and JSMA. The results showed that classifier models with more than 90% accuracy experienced a performance drop to 75-55% when exposed to maliciously crafted samples. From a different perspective, in [33], the impact of adversarial samples on ML/DL models for user identification based on motion sensors was evaluated, achieving nearly 100% attack success rates with FGSM, JSMA, DeepFool, and Boundary Attacks. In [34], it was shown that GAN-based attacks had even more significant effects in the user identification context. In [14] and [35], a review of adversarial attacks in ML solutions applied in network security was conducted, demonstrating the substantial impact of adversarial attacks on ML-based security systems and underscoring the need for further research on attack and defense methods in this area.

The novelty of the proposed approach lies in its exploration of uncharted territory within the field of cybersecurity. Several key aspects set this work apart from previous research:

- Combining context and ML/DL attacks: At first, unlike previous studies, context-based attacks were combined with ML/DL-focused attacks. This fusion of attack methodologies has not been extensively investigated in the past. This focus opens new avenues for understanding and mitigating threats that arise from both contextual factors and advanced ML/DL techniques.

- Untapped potential of defense mechanisms: Furthermore, while ML and DL-focused attacks have been examined in the context of device or user identification, previous studies did not fully exploit the potential of existing defense mechanisms. This study recognized this oversight and actively explored the benefits and effectiveness of these defense strategies.

- Addressing a literature gap: This study serves as a pioneering contribution to the field, offering valuable insight into the intricate interplay between attack and defense techniques within the specific context of hardware-based individual device identification and LSTM/ML-based methods. By addressing this gap, this study enriches the understanding of security challenges in these domains and paves the way for more robust and effective security solutions.

III. THE PROPOSED ARCHITECTURE

A. Threat Model

The threat model centers on the phase where the identification solution is already trained and operational, thus exclusively focusing on threats that impact device evaluation. In this context, potential attackers can target either the hardware that generates the data or the LSTM/ML models responsible for data assessment. Within this framework, the following threats are identified:

- Fingerprint eavesdropping and hijacking [36]: An adversary could intercept the data that constitute a fingerprint, whether during in-device data collection, communication, or processing (on a server or the device itself). The data obtained could then be used on another device to impersonate the identity of the original device. This threat assumes limited knowledge of the fingerprint generation process, as well as of the functions and components used during the process.

- Fingerprint forgery [37]: Given that the components and frequencies of the device are public knowledge, an attacker who knows the functions used to generate fingerprints might attempt to craft a new fingerprint that closely resembles that of a legitimate device. This threat might involve a trial-and-error or brute-force approach, necessitating an in-depth understanding of the fingerprint generation process and the values comprising the fingerprint.

- Context modification [38]: As fingerprints are constructed from data collected during the execution of specific software tasks, an attacker might attempt to alter the conditions under which the fingerprint is generated. This could lead to the failure to recognize a genuine device or result in forged fingerprints emulating another device. Context manipulation can take various forms, such as increasing the device temperature (using external tools or intensive hardware usage) or introducing software that introduces kernel interruptions into the fingerprint collection process. Efforts must be made to isolate the fingerprint collection program from these interactions as much as possible.

- ML/DL evaluation evasion [39]: In LSTM/ML-based solutions, an attacker possessing sufficient access to the evaluation model could craft malicious data samples to deceive the LSTM/ML solution. These samples might impersonate a specific device through trial and error or targeted attacks, as mentioned above.

In light of these considerations, a robust individual device identification solution must thoroughly account for and evaluate these identified threats. Doing so is essential to ensure accurate operation and resilience against potential attacks. The LSTM/ML framework is run for hardware-based individual device identification, providing the foundational results that will underpin subsequent analysis of attack and defense techniques.

B. Data Collection and Preprocessing

Within the realm of hardware-driven individual device recognition, it becomes crucial to oversee the variations occurring within the device's chips, with the intent of subsequent assessment. Previous studies contrasted components via different crystal oscillators or fundamental frequencies, leveraging the ability to detect discrepancies in component behavior originating directly from the device. Establishing the foundation for individual device identification requires the creation of a dataset comprising metrics linked to specific hardware components within devices. This dataset, labeled LwHbench, was further expanded in [16]. In pursuit of this goal, performance metrics that include GPU, CPU, Storage, and Memory were amassed from 45 distinct RPi devices representing various models over 100 days. An array of functions was executed across these components, with other hardware components operating at distinct frequencies to facilitate performance measurement. These functions were related to device core temperature, timestamp, elapsed GPU cycles during sleep, times taken by CPU, memory, storage, and others. The dataset consists of a set of samples as described in Table I. Countermeasures were taken to mitigate noise from other processes on the devices, including fixed component frequency, kernel-level priority, execution on an isolated CPU core, and randomized memory address deactivation. In addition, the dataset was built according to the variation of the temperature conditions, facilitating analysis of the contextual impact on component performance.

TABLE I. LWHBENCH DATASET FEATURES

Hardware Model	Number of boards	Number of samples
RPi 1B+	10	505,584
RPi4	15	784,095
RPi3	10	547,800
RPiZero	10	548,647

Following the approach outlined in [6], feature extraction using sliding windows was performed in each device. This involved extracting statistical features such as median, average, maximum, minimum, and summation. The rationale behind this lies in the overlap in the distribution of raw feature values across devices due to limited variability in component performance. Thus, statistical metrics, such as median and average, help to distinguish partially overlapping distributions.

For this step, only a subset of available raw features was chosen, as having fewer features helps to streamline the training of ML/DL models. A total of 440 features were extracted from each device's dataset, composed of many operations such as sleeping time, string hashing, urandom, matrix mul, matrix sum, list creation, memory reserve, CSV read, storage read, and storage write. In addition to sliding windows, a direct assessment of raw data vectors without the aforementioned sliding window processing was undertaken. This approach was guided by the belief that a vast dataset of raw values could yield favorable results in DL models, which can gain internal insights from the data. In this scheme, only timestamp and temperature features were filtered and the remaining values (totaling 215) were used as features for the models. Furthermore, a time series-based assessment was performed, in which samples were concatenated into groups of 10 vectors. This grouping method allows the application of time series DL models like LSTM and 1D-CNN [40], which are adept at extracting intricate trends that can yield superior results than the isolated processing and assessment of individual samples.

Mitigating noise in the LwHBench dataset is essential to ensure the accuracy of data analysis and identification tasks. Several countermeasures were used:

- Isolation and Resource Allocation: Allocate dedicated resources to target processes to reduce interference from background tasks.
- Real-Time Systems: Use real-time operating systems to prioritize critical tasks and prevent interruptions during data collection.
- Data Filtering and Preprocessing: Apply data filtering and preprocessing techniques, such as signal processing and outlier removal, to extract relevant information and eliminate noise.
- Data Averaging: Reduce the impact of transient noise by averaging multiple measurements over time to obtain a more stable dataset.

C. Architecture of LSTM-MLP

This study used an LSTM-MLP neural network to categorize performance data acquired from the devices. This architecture has shown robust capabilities in various time series contexts, such as identifying human activities [41], predicting gold prices [42], and analyzing DNA protein binding [43]. The structure of the neural network merges the LSTM and MLP layers, facilitating the extraction of patterns from the input sequences. The significant benefit of this approach is the combination of recurring patterns obtained through the LSTM layer's memory capabilities and the spatial patterns derived by the MLP layer's utilization of kernels on neighboring features, resulting in the generation of more intricate patterns. Figure 1 provides an overview of the LSTM-MLP architecture to distinguish sanctioned and unsanctioned devices. In the training phase, the binary cross-entropy loss function is applied to samples as follows:

$$L = -\frac{1}{n}\sum_{s}[R_l \log(p) + (1 - R_l)\log(1 - p)] \qquad (1)$$

where s is the number of features in the vector, R_l is the real label of the identification attempt, and p is the probability of the predicted model.

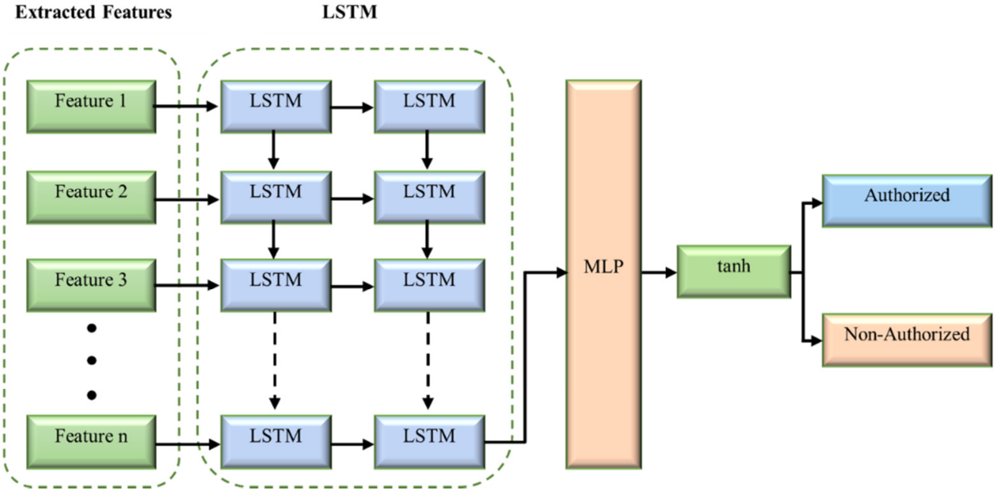


Fig. 1. The proposed LSTM-MLP architecture.

IV. EXPERIMENTATION AND RESULTS

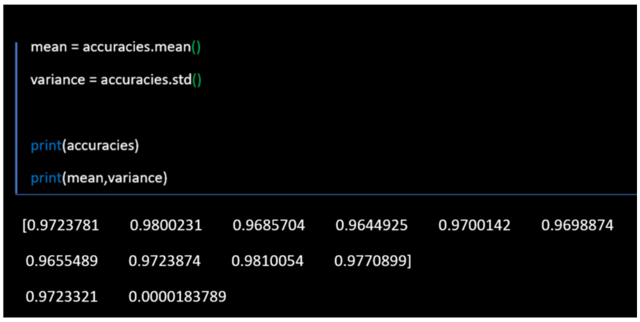
After applying the two data preprocessing approaches to generate the two datasets, one encompassing raw values and the other incorporating sliding-window-based features, device identification experiments were carried out to compare the proposed LSTM-MLP and prevalent ML/DL classification models. In [6], ML classifiers were directly used, utilizing statistical features related to CPU and GPU. In addition, the LSTM and MLP networks were evaluated for time series approaches. Additionally, a more intricate multi-input network, fusing one LSTM and one MLP input layer, was implemented for comparison. These experiments were carried out on a server equipped with an Intel Xeon Silver 4310 CPU and an NVIDIA

A100 GPU. Table II shows the algorithms and hyperparameters explored. Furthermore, for algorithms that require data normalization, the Quantile Transformer [44] was applied due to variations in data distribution between different device models based on hardware capabilities. Training and cross-validation used 80% of the data, while testing used the remaining 20%. To avoid potential order-related correlations, the train/test split was conducted without vector shuffling.

TABLE II. CLASSIFICATION ALGORITHMS AND HYPERPARAMETERS

Model	Hyperparameters
Naive Bayes	No hyperparameter tunning required
k-NN	K ∈ [3, 20]
SVM	C ∈ [0.01, 100], gamma ∈ [0.001, 10], kernel ∈{'rbf', linear',' sigmoid',' poly'}
AdaBoost	n_estimators ∈ [10, 100]
XGBoost	lr ∈ [0.01, 0.3], max_depth ∈ [3,15], min_child_weight ∈ [1, 7], gamma ∈ [0, 0.5], colsample_bytree ∈ [0.3, 0.7]
Decision Tree	max_depth ∈ [None, 5, 10, 15, 20], min_samples_split ∈ [2, 3, 4, 5]
Random Forest	number_of_trees ∈ [50, 1000], max_depth ∈ [None, 5, 10, 15, 20], min_samples_split ∈ [2, 3, 4, 5]
MLP	n_layers ∈ [1, 3], neurons_layer ∈ [100, 500], batch_size ∈ [32, 64, 128, 256, 512] activation = relu, optimizer = [SGD, adam, adamax]
1D-CNN	filters = [16, 32, 64, 128], kernel_size = [3, 5, 7], n_layers = [1, 2, 3], optimizer = [SGD, adam, adamax]
LSTM	neurons = [10, 100], n_layers = [1, 2, 3], optimizer [SGD, adam, adamar]
Multi_1DCNN LSTM	input_layers = [2, 3], cnn_filters = [16, 32, 64, 128], cnn_kernel_size = [3, 5, 7], 1stm_neurons = [10, 100], n_layers = [1, 2, 3], optimizer = [SGD, adam, adamax]

The MLP model was iterated across various epoch counts (50, 100, 150, and 250) accompanied by batch sizes ranging from 300 to 500. After evaluation, it was determined that the most favorable configuration consisted of an epoch count of 150 paired with a batch size of 500. During training, ten rounds of cross-validation tests were carried out and Figure 2 presents the results, showcasing an average accuracy of 97.23%, indicating its notable effectiveness.



```
mean = accuracies.mean()

variance = accuracies.std()


print(accuracies)

print(mean,variance)
```

```
[0.9723781    0.9800231    0.9685704    0.9644925    0.9700142    0.9698874
 0.9655489    0.9723874    0.9810054    0.9770899]
 0.9723321    0.0000183789
```

Fig. 2. Accuracy results.

Table III presents a comparison between the proposed LSTM/MLP and other artificial intelligence models. The LSTM-MLP model had the most robust performance, achieving approximately 97% accuracy when using raw data features, as shown in Figure 3.

TABLE III. COMPARISON BETWEEN MODELS

Model	Accuracy	Precision	Recall	F1-score
SVM	78.3%	79.5%	78.2%	78.4%
k-NN	45.2%	46.7%	45.2%	44.7%
Naïve Bayes	45.6%	47.3%	45.6%	44.7%
XGBoost [6]	90.5%	91.7%	90.5%	90.8%
AdaBoost [45]	7%	0.6%	7%	1.1%
Decision Tree [46]	78.1%	78.9%	78.2%	78.3%
Random Forest [47]	85.4%	86.6%	85.4%	85.7%
MLP [48]	88.9%	89.6%	88.8%	88.9%
LSTM [49]	93.4%	94.3%	93.4%	93.4%
1D-CNN [50]	94.2%	94.5%	94.2%	94.2%
1D-CNN-LSTM [51]	95.3%	95.5%	95.3%	95.3%
LSTM-MLP	**97.2%**	**97.6%**	**97.5%**	**97.5%**

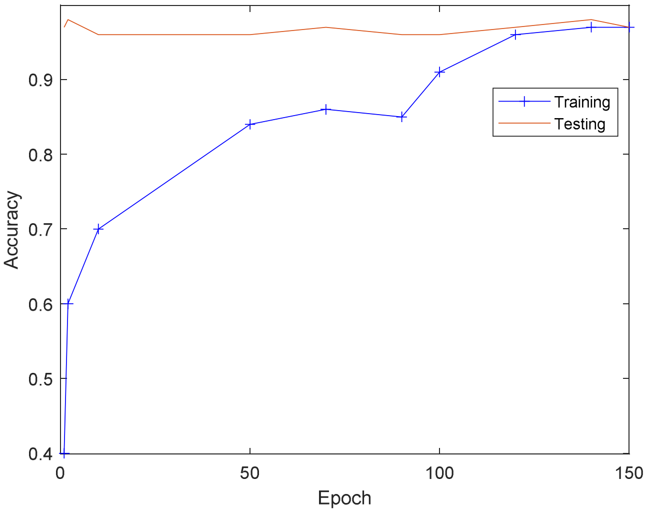


Fig. 3. MLP trained-test accuracy.

Metrics such as accuracy, precision, recall, and F1-score assess the performance of classification models. Accuracy calculates the proportion of accurately predicted instances compared to the total instances within a dataset, providing a holistic understanding of the model's performance in accurately categorizing data points. Precision focuses on the number of accurate positive predictions compared to the total instances predicted as positive, denoting the percentage of positive predictions that were genuinely accurate. Precision becomes valuable when there is a significant consequence for making false positive predictions. Recall evaluates the accurate positive predictions for the total actual positive instances, showcasing the model's capability to correctly recognize all occurrences of a specific class. The F1-score constitutes the harmonic average of precision and recall, establishing an equilibrium between them considering both false positives and false negatives. The F1-score proves especially beneficial when aiming for a trade-off between precision and recall, particularly in scenarios characterized by an imbalanced distribution of classes. The LSTM-MLP provided higher results by achieving approximately 97.2% accuracy, 97.6% precision, 97.5% recall, and 97.5% F1-score.

V. DISCUSSION

The proposed LSTM-MLP model introduced significant advantages that promise to revolutionize the way to safeguard

IoT ecosystems and authenticate individual devices. Foremost among its merits is its ability to significantly enhance security in IoT environments. By shifting the paradigm from traditional static identifiers, such as MAC addresses, to dynamic hardware performance behavior, the LSTM-MLP model thwarts attackers' attempts to impersonate or clone devices. This innovative approach creates formidable barriers against malicious incursions, reinforcing the overall security infrastructure of IoT ecosystems. Moreover, the model masterfully addresses the pervasive vulnerability of spoofing, a perennial concern in IoT security. Static identifiers are notoriously susceptible to manipulation and imitation, whereas the LSTM-MLP model, rooted in behavior-based attributes, resiliently resists such attempts and mitigates a major security loophole in IoT environments.

The LSTM-MLP's proficiency in anomaly detection is another hallmark of its utility. Protection of IoT devices involves vigilant monitoring for deviations from established behavior patterns, as they could signal security breaches or device malfunctions. The model's ability to capture long-term dependencies within device behavior data proves invaluable, as it excels in identifying unusual activities, facilitates the early detection of potential threats, and enables swift responses to security incidents. This adaptability extends to the model's ability to accommodate the inherent variability among IoT devices. The IoT encompasses a multitude of devices, each with its unique hardware components and performance characteristics. The LSTM-MLP model exhibits remarkable adaptability, learning to differentiate between individual devices regardless of their idiosyncrasies. This adaptability positions it as a versatile and reliable solution for heterogeneous IoT environments. Furthermore, the model's resilience to environmental fluctuations is notable. IoT devices operate in diverse and dynamic settings. The proficiency of the LSTM-MLP model in processing sequential data and adapting to evolving behavior patterns makes it suitable for such challenging scenarios, as it remains strong in the face of environmental changes and maintains the integrity of device identification and security measures.

False positives in security alerts have been a persistent headache for IoT security practitioners. However, the LSTM-MLP model reduces the incidence of false alarms, as it distinguishes between normal variations in device behavior and genuine security threats with a high degree of accuracy. This accuracy translates into more actionable and reliable alerts, allowing security teams to respond effectively to real threats. Scalability is a quintessential requirement for IoT, which often involves vast networks of numerous devices. The LSTM-MLP model excels in this regard, demonstrating the ability to process and identify devices in real-time, even at the scale of IoT ecosystems consisting of thousands or even millions of devices. This scalability is crucial for the effective implementation of security measures in expansive IoT deployments. The versatility of the LSTM-MLP model is further exemplified by its applicability in a wide spectrum of IoT use cases. Whether it is securing smart homes, industrial IoT systems, healthcare applications, or other domains, the model proves its mettle in protecting device integrity and fortifying security.

A key attribute that elevates the LSTM-MLP model is its capacity for continuous learning. In the ever-evolving landscape of IoT security, threat landscapes morph and device behavior patterns evolve. Its ability to adapt and refine its accuracy over time is a strategic advantage, as it ensures that IoT security measures remain effective, even as threats grow in sophistication and diversity. Lastly, the model significantly reduces reliance on centralized servers, a potential weak point in IoT security architectures. Although some approaches heavily depend on centralized infrastructure for authentication and identification, the LSTM-MLP model can operate locally on devices. This decentralization minimizes the risk associated with single points of failure and network vulnerabilities, reinforcing the robustness of IoT security.

VI. CONCLUSIONS AND FUTURE WORK

The surge in IoT device deployment has spurred the creation of novel device identification solutions based on hardware behavior and ML/DL processing. However, these solutions face challenges from adversarial attacks that aim to undermine their efficiency. This study delved into the assessment of hardware behavior-based device identification performance. The LwHBench dataset, comprising samples from 45 RPi devices operating on identical software images, was harnessed to train ML/DL classifiers entrusted with individual device identification. This study proposed an artificial intelligence model that combined LSTM and MLP. The experimental results showed an average F1-score of 97.5% while successfully identifying all devices by setting a threshold at a TPR of 97.2%, outperforming other methods. Future work involves exploring adversarial attack and defense techniques, including those grounded in generative models, to comprehensively improve the robustness of this solution. Additionally, the research roadmap involves experimenting with fully distributed model generation and harnessing federated learning to circumvent data sharing and centralization challenges.

REFERENCES

[1] A. Ayub Khan, A. A. Laghari, Z. A. Shaikh, Z. Dacko-Pikiewicz, and S. Kot, "Internet of Things (IoT) Security With Blockchain Technology: A State-of-the-Art Review," *IEEE Access*, vol. 10, pp. 122679–122695, 2022, https://doi.org/10.1109/ACCESS.2022.3223370.

[2] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020, https://doi.org/10.1109/ACCESS.2020.2970118.

[3] A. E. Omolara *et al.*, "The internet of things security: A survey encompassing unexplored areas and new insights," *Computers & Security*, vol. 112, Jan. 2022, Art. no. 102494, https://doi.org/10.1016/j.cose.2021.102494.

[4] S. Halder and T. Newe, "Radio fingerprinting for anomaly detection using federated learning in LoRa-enabled Industrial Internet of Things," *Future Generation Computer Systems*, vol. 143, pp. 322–336, Jun. 2023, https://doi.org/10.1016/j.future.2023.01.021.

[5] I. Sanchez-Rola, I. Santos, and D. Balzarotti, "Clock Around the Clock: Time-Based Device Fingerprinting," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, ON, Canada, Jul. 2018, pp. 1502–1514, https://doi.org/10.1145/3243734.3243796.

[6] P. M. Sánchez Sánchez, J. M. Jorquera Valero, A. Huertas Celdrán, G. Bovet, M. Gil Pérez, and G. M. Pérez, "A methodology to identify

identical single-board computers based on hardware behavior fingerprinting," *Journal of Network and Computer Applications*, vol. 212, Mar. 2023, Art. no. 103579, https://doi.org/10.1016/j.jnca.2022.103579.

[7] Y. Li, Y. Zuo, H. Song, and Z. Lv, "Deep Learning in Security of Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22133–22146, Aug. 2022, https://doi.org/10.1109/JIOT.2021.3106898.

[8] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine Learning for the Detection and Identification of Internet of Things Devices: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 298–320, Jan. 2022, https://doi.org/10.1109/JIOT.2021.3099028.

[9] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, https://doi.org/10.1007/s10462-020-09825-6.

[10] K. Sadeghi, A. Banerjee, and S. K. S. Gupta, "A System-Driven Taxonomy of Attacks and Defenses in Adversarial Machine Learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 450–467, Dec. 2020, https://doi.org/10.1109/TETCI.2020.2968933.

[11] Z. Tian, L. Cui, J. Liang, and S. Yu, "A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning," *ACM Computing Surveys*, vol. 55, no. 8, Sep. 2022, Art. no. 166-166, https://doi.org/10.1145/3551636.

[12] M. Aprilpyone, Y. Kinoshita, and H. Kiya, "Adversarial Robustness by One Bit Double Quantization for Visual Classification," *IEEE Access*, vol. 7, pp. 177932–177943, 2019, https://doi.org/10.1109/ACCESS.2019.2958358.

[13] M. Al-Rubaie and J. M. Chang, "Privacy-Preserving Machine Learning: Threats and Solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, Mar. 2019, https://doi.org/10.1109/MSEC.2018.2888775.

[14] K. He, D. D. Kim, and M. R. Asghar, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023, https://doi.org/10.1109/COMST.2022.3233793.

[15] T. Laor *et al.*, "DRAWNAPART: A Device Identification Technique based on Remote GPU Fingerprinting," in *Proceedings 2022 Network and Distributed System Security Symposium*, 2022, https://doi.org/10.14722/ndss.2022.24093.

[16] P. M. Sánchez Sánchez, J. M. Jorquera Valero, A. Huertas Celdrán, G. Bovet, M. Gil Pérez, and G. Martínez Pérez, "LwHBench: A low-level hardware component benchmark and dataset for Single Board Computers," *Internet of Things*, vol. 22, Jul. 2023, Art. no. 100764, https://doi.org/10.1016/j.iot.2023.100764.

[17] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," *Computer Networks*, vol. 183, Dec. 2020, Art. no. 107593, https://doi.org/10.1016/j.comnet.2020.107593.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples." arXiv, Mar. 20, 2015, https://doi.org/10.48550/arXiv.1412.6572.

[19] J. Wang, "Adversarial Examples in Physical World," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Montreal, Canada, Aug. 2021, pp. 4925–4926, https://doi.org/10.24963/ijcai.2021/694.

[20] Y. Dong *et al.*, "Boosting Adversarial Attacks with Momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, Jun. 2018, pp. 9185–9193, https://doi.org/10.1109/CVPR.2018.00957.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks." arXiv, Sep. 04, 2019, https://doi.org/10.48550/arXiv.1706.06083.

[22] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2574–2582, https://doi.org/10.1109/CVPR.2016.282.

[23] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbruecken, Germany, Mar. 2016, pp. 372–387, https://doi.org/10.1109/EuroSP.2016.36.

[24] K. J. Waldron, S. L. Wang, and S. J. Bolin, "A Study of the Jacobian Matrix of Serial Manipulators," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 107, no. 2, pp. 230–237, Jun. 1985, https://doi.org/10.1115/1.3258714.

[25] W. Brendel, J. Rauber, and M. Bethge, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models." arXiv, Feb. 16, 2018, https://doi.org/10.48550/arXiv.1712.04248.

[26] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, Feb. 2017, pp. 39–57, https://doi.org/10.1109/SP.2017.49.

[27] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain," *ACM Computing Surveys*, vol. 54, no. 5, Feb. 2021, Art. no. 108, https://doi.org/10.1145/3453158.

[28] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training." arXiv, Jan. 12, 2020, https://doi.org/10.48550/arXiv.2001.03994.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network." arXiv, Mar. 09, 2015, https://doi.org/10.48550/arXiv.1503.02531.

[30] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, May 2016, pp. 582–597, https://doi.org/10.1109/SP.2016.41.

[31] Z. Bao, Y. Lin, S. Zhang, Z. Li, and S. Mao, "Threat of Adversarial Attacks on DL-Based IoT Device Identification," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 9012–9024, Jun. 2022, https://doi.org/10.1109/JIOT.2021.3120197.

[32] A. Namvar, C. Thapa, S. S. Kanhere, and S. Camtepe, "Evaluating the Security of Machine Learning Based IoT Device Identification Systems Against Adversarial Examples," in *Service-Oriented Computing*, 2021, pp. 800–810, https://doi.org/10.1007/978-3-030-91431-8_57.

[33] C. Benegui and R. T. Ionescu, "Adversarial Attacks on Deep Learning Systems for User Identification Based on Motion Sensors," in *Neural Information Processing*, Bangkok, Thailand, 2020, pp. 752–761, https://doi.org/10.1007/978-3-030-63823-8_85.

[34] N. Pourshahrokhi, M. Smith-Creasey, M. Ghassemian, and S. Kouchaki, "Generative adversarial attacks on motion-based continuous authentication schemes," in *2021 14th International Conference on Security of Information and Networks (SIN)*, Edinburgh, United Kingdom, Sep. 2021, vol. 1, pp. 1–6, https://doi.org/10.1109/SIN54109.2021.9699365.

[35] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems," *Digital Threats: Research and Practice*, vol. 3, no. 3, Oct. 2022, Art. no. 31, https://doi.org/10.1145/3469659.

[36] Y. Chen and Y. He, "BrutePrint: Expose Smartphone Fingerprint Authentication to Brute-force Attack." arXiv, May 18, 2023, https://doi.org/10.48550/arXiv.2305.10791.

[37] H. Miao, Y. Guo, and Y. Wang, "RFDforFin: Robust Deep Forgery Detection for GAN-generated Fingerprint Images." arXiv, Sep. 13, 2023, https://doi.org/10.48550/arXiv.2308.09285.

[38] Z. X. Li, Y. J. Li, Y. W. Liu, C. Liu, and N. X. Zhou, "K-CTIAA: Automatic Analysis of Cyber Threat Intelligence Based on a Knowledge Graph," *Symmetry*, vol. 15, no. 2, Feb. 2023, Art. no. 337, https://doi.org/10.3390/sym15020337.

[39] M. A. Gill, N. Ahmad, M. Khan, F. Asghar, and A. Rasool, "Cyber Attacks Detection Through Machine Learning in Banking," *Bulletin of Business and Economics (BBE)*, vol. 12, no. 2, pp. 34–45, Aug. 2023, https://doi.org/10.5281/zenodo.8310116.

[40] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, Apr. 2021, Art. no. 107398, https://doi.org/10.1016/j.ymssp.2020.107398.

[41] S. Zhang *et al.*, "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances," *Sensors*, vol. 22, no. 4, Jan. 2022, Art. no. 1476, https://doi.org/10.3390/s22041476.

[42] Z. He, J. Zhou, H.-N. Dai, and H. Wang, "Gold Price Forecast Based on LSTM-CNN Model," *in 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, Fukuoka, Japan, Dec. 2019, pp. 1046–1053, https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00188.

[43] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome," *Bioinformatics*, vol. 37, no. 15, pp. 2112–2120, Aug. 2021, https://doi.org/10.1093/bioinformatics/btab083.

[44] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies*, vol. 9, no. 3, Sep. 2021, Art. no. 52, https://doi.org/10.3390/technologies9030052.

[45] B. M. M. AlShahrani and E. Al, "Classification of Cyber-Attack using Adaboost Regression Classifier and Securing the Network," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 1215–1223, Apr. 2021, https://doi.org/10.17762/turcomat.v12i10.4314.

[46] Q. H. Vu, D. Ruta, and L. Cen, "Gradient boosting decision trees for cyber security threats detection based on network events logs," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, Sep. 2019, pp. 5921–5928, https://doi.org/10.1109/BigData47090.2019.9006061.

[47] M. Choubisa, R. Doshi, N. Khatri, and K. Kant Hiran, "A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security," in *2022 International Conference on IoT and Blockchain Technology (ICIBT)*, Ranchi, India, Feb. 2022, pp. 1–5, https://doi.org/10.1109/ICIBT52874.2022.9807766.

[48] T. T. Teoh, G. Chiew, E. J. Franco, P. C. Ng, M. P. Benjamin, and Y. J. Goh, "Anomaly detection in cyber security attacks on networks using MLP deep learning," in *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, Shah Alam, Malaysia, Jul. 2018, pp. 1–5, https://doi.org/10.1109/ICSCEE.2018.8538395.

[49] H. Gasmi, J. Laval, and A. Bouras, "Information Extraction of Cybersecurity Concepts: An LSTM Approach," *Applied Sciences*, vol. 9, no. 19, Jan. 2019, Art. no. 3945, https://doi.org/10.3390/app9193945.

[50] A. Khan and C. Cotton, "Detecting Attacks on IoT Devices using Featureless 1D-CNN," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, Rhodes, Greece, Jul. 2021, pp. 461–466, https://doi.org/10.1109/CSR51186.2021.9527910.

[51] M. Al-Khafajiy, G. Al-Tameemi, and T. Baker, "DDoS-FOCUS: A Distributed DoS Attacks Mitigation using Deep Learning Approach for a Secure IoT Network," in *2023 IEEE International Conference on Edge Computing and Communications (EDGE)*, Chicago, IL, USA, Jul. 2023, pp. 393–399, https://doi.org/10.1109/EDGE60047.2023.00062.