Ensemble Modelling based Attestation Framework Design for IoT Network Security

Mini TT
ISG Engineering and Development
Dell Technologies
Bengaluru, India
mini.tt@dell.com

Santrupti Behera
Dept. of EEE
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20211724@hyderabad.bits-pilani.ac.in

Meka Srikar Reddy
Dept. of EEE
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20201652@hyderabad.bits-pilani.ac.in

name of author
line 2: *dept. name of organization*
(of Affiliation)
line 3: *name of organization*
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

***Abstract*— Attestation entails presenting verifiable evidence to an evaluator to substantiate claims regarding a target's characteristics. Devices that handle sensitive data go through multiple stages in their lifecycle, with their computing capabilities varying based on their type, such as from wearables to Industrial IoT devices or computing servers. These devices are prone to threats like edge-based attacks, DoS attacks, counterfeiting and firmware alteration. Attestation ensures that the firmware and configuration are reliable, verifying that the hardware is genuine, the firmware and configurations are untouched, and the assembly of composite devices remains unaltered. A review of current attestation solutions and their application across the device lifecycle indicate significant gaps, especially a lack of comprehensive lifecycle solutions and support for large-scale networks. This paper proposes attestation solutions based on optimized ensemble learning and neural networks to bridge the existing gaps, enhancing security in the device lifecycle.**

Keywords—Attestation, IoT Security, Device Verification, Edge-based IoT attacks

I. Introduction

Attestation is a mechanism used to verify the authenticity and integrity of the hardware and firmware of the device by a relying party. Attestation ensures that the device is genuine, untampered, and performing its operations as expected. The tampering could be affecting the hardware, firmware or configuration of the device. While attestation based on the device creator based information such as hardware identify certificate, and firmware measurements are described in multiple prior studies, continuous device attestation based on the behavior of the device to according to its operational environment creates security challenges for the device. Some of the examples of security challenges in the runtime environment are injection based attacks, ransomware, and cross site scripting.

A typical attestation architecture is shown in Figure 1. The architecture is as per the proposal by IETF [30].

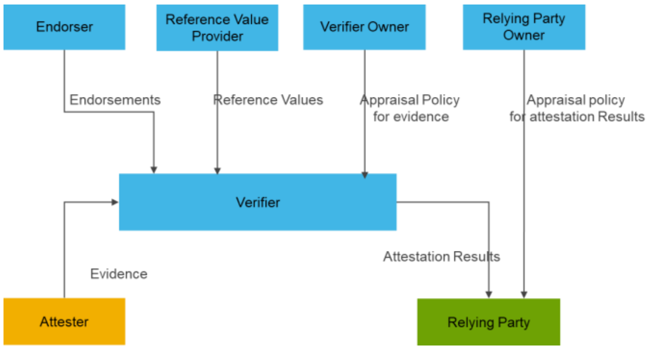


Figure 1: Attestation Architecture

The Attester is the device whose trustworthiness has to be established. The Attester produce evidence that is conveyed to a Verifier.

The Verifier uses Evidence, any Reference Values from Reference Value Providers, and any Endorsements from Endorsers by applying an Appraisal Policy for Evidence to assess the trustworthiness of the Attester. The Verifier generates Attestation Results for use by Relying Parties.

The Appraisal Policy for Evidence might be obtained from the Verifier Owner via some protocol mechanism, or might be configured into the Verifier by the Verifier Owner, or might be programmed into the Verifier, or might be obtained via some other mechanism.

The Relying Party uses Attestation Results by applying its own appraisal policy to make application-specific decisions, such as authorization decisions. This procedure is called the appraisal of Attestation Results.

The Appraisal Policy for Attestation Results might be obtained from the Relying Party Owner via some protocol mechanism, or might be configured into the Relying Party by the Relying Party Owner, or might be programmed into the Relying Party, or might be obtained via some other mechanism.

Runtime attestation based on in-field attack scenarios require mechanisms based on attack patterns. This paper describes a machine learning based attestation solution where models are trained on feature engineered datasets covering potential run time attacks.

II. Literature review

A. *Traditional Attestation Methods*

The principles of remote attestation were discussed in **Error! Reference source not found.**. This paper introduced the fundamental principles of attestation, including the terminologies related to attestation and the fundamental principles.

The remote attestation framework for IoT devices is discussed in **Error! Reference source not found.**. SPDM **Error! Reference source not found.** is a protocol by DMTF for providing attestation evidence. A study on securing hard drives with SPDM is discussed in **Error! Reference source not found.**. Though this study is focused on storage devices, similar concepts are applicable to other component devices which form part of a larger composite assembly. A proposal for post quantum design of SPDM for device authentication and key establishment is presented in **Error! Reference source not found.**. DMTF standards forum has not addressed post-quantum challenges with SPDM protocol. A formal analysis of the security of SPDM protocol message exchanges is presented in **Error! Reference source not found.**. This analysis provides visibility into the security of the message exchanges, and there is no other significant analysis of attestation at a system level.

A detailed study on secure compute enclave-based attestation is covered in **Error! Reference source not found.**. Various aspects of attestation, such as identities, models of attestation, and composite device attestation, are discussed in detail in this paper. Another study on remote attestation schemes as a taxonomy and review is presented in **Error! Reference source not found.**. Other literature reviews on remote attestation may be found in **Error! Reference source not found.**. The literature primarily focuses on component-level attestation. Attestation at a system level is an open topic of research which needs further study.

Architecture for remote attestation is worked on by the IETF Remote Attestation Task Group **Error! Reference source not found.**. Defining the actors involved in the attestation architecture and the data flow paths is a major contribution to this architecture. The definition of data flows is limited to conceptual messages without defining the data structures and schema for the message exchanges. Another attestation architecture is by TCG **Error! Reference source not found.**, which also considers low-cost attestation based on DICE.

A device in its lifecycle can be compromised at any point in the supply chain, operation and decommissioning. Considering this possibility, attestation frameworks need to consider the device lifecycle and onboarding mechanism. Various onboarding solutions are proposed by different standards groups considering the needs of the industry segment. OPCUA Device boarding **Error! Reference source not found.** is targeted for industrial device onboarding to an operational network. Bootstrapping Remote Secure Key Infrastructure **Error! Reference source not found.** provides solutions for device onboarding for scenarios considering different connectivity options. These include internet-connected devices, disconnected scenarios and makes suitable security assumptions. Secure Zero Touch Provisioning **Error! Reference source not found.** considers provisioning the operating system and configurations apart from hardware provisioning. A provisioning solution specifically meant for IoT devices is FIDO Device Onboarding **Error! Reference source not found.**. While these onboarding solutions consider hardware identities, they do not consider attestation to ensure that the firmware and configuration are trustworthy.

B. *Machine Learning Based Attestation*

An attestation method based on memory traces is described in [2]. This technique collects memory traces from the device main memory and converts them to grayscale images. Machine learning based classifiers are applied on these images to detect any tampering with the device firmware.

Traditional firmware attestation methods often depend on having an authorized copy of the firmware, which is frequently unavailable due to its status as the manufacturer's intellectual property (IP). Existing machine learning (ML) [2] classifiers are based on analysis of memory dumps from IoT devices, increasing the computational complexity of remote verifiers assessing the integrity of the devices' internal state.

III. Proposed solution

This paper proposes a novel attestation framework for Malicious IoT device and Malware attack-type detection in real-time, obviating the need for a legitimate copy of the original firmware. This framework is based on a Bayesian optimized Maximum Voting Classification ensemble model with Random Forest, Logistic Regression, Extreme Gradient Boosting and Gaussian Naïve Bayes as base models for binary classification. Furthermore, multilabel and multiclass classification is based on Stack ensemble model with Decision Trees, K-Nearest Neighbors (KNN), Support Vector Machines (SVM and Random Forest as base models with AdaBoost meta-learner.

Both machine learning models were trained on a feature-engineered realistic cyber security dataset that covers Ransomware, Man in the middle and SQL Injection attacks, to name a few out of 14 other attacks detected in the IoT and Industrial IoT Perception Layer, gathered from 10 different types of IoT devices. The attack distribution is shown in Figure 2.

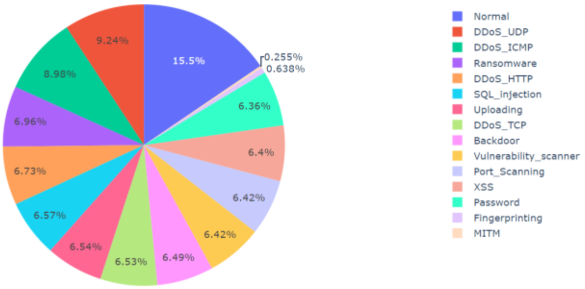


Figure 2: Distribution of attacks in the IoT and IIoT Perception Layer.

Based on the data analysis of testbed and evaluating the performance of the proposed approaches, the framework promises an accuracy of 93% with 92% precision for detection of compromised IoT device and attack type.

IV. Methodology and Materials

A. Data Analysis and Feature Engineering

Exploratory data analysis (EDA) is foundation to building robust intrusion detection systems in IoT environment. Data cleaning for null and duplicate values, feature scaling using Standard and Min-Max Scaling and Label encoding of categorical values, were performed on the dataset.

Finally, dataset was partitioned into training, validation and testing subsets followed by Synthetic Minority Oversampling (SMOTE) to address imbalanced classes. Features of the dataset were extracted and correlation coefficients between features were established to identify potential relationships. Figure 3 visualizes 15 attributes using a heatmap highlighting redundancy or important feature interactions extracted from dataset.

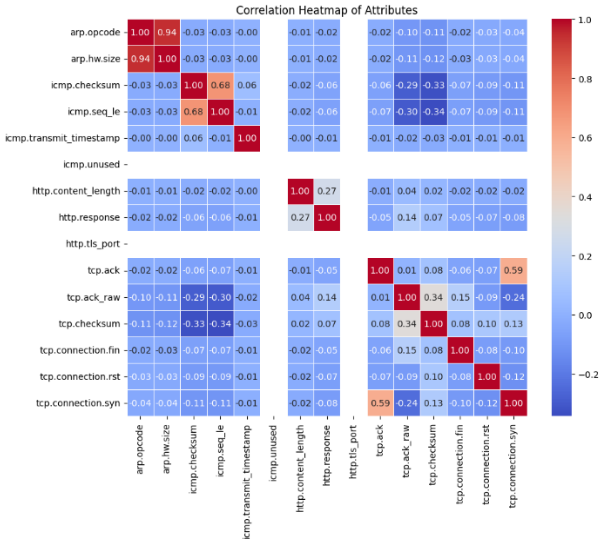


Figure 3: Correlation Heatmap of Features.

Further, dimensions of the dataset were reduced using Fisher's Linear Discriminant Analysis to reduce the feature space while retaining essential information and enhancing model performance. The feature space scaled, sampled and dimensionally reduced was then used to train the base models of maximum voting ensemble classifier and stack-based ensemble model and comparing its performance against that of Convolutional Neural Network (CNN) model with Adam optimization.

B. Model Training

Ensemble Models such as Maximum Voting Classifiers are typically effective in heterogeneous IoT environment, where they enhance generalization and mitigate overfitting risks [8]. With *Soft Voting*, probability estimates for each class are aggregated from each base classifier with the ensemble predicting the class with the highest average probability.

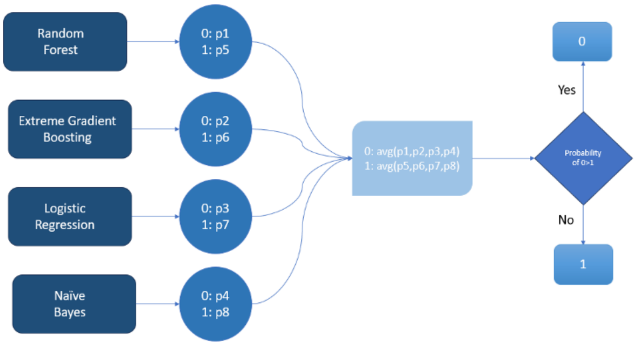


Figure 4: Diagrammatic representation of maximum voting ensemble model with Soft Voting.

Stacking ensemble technique leverages the strengths of multiple machine learning models [9]. This approach involves training each of the base learners on the dataset and then combining their predictions using a *Meta-learner* to make the final prediction efficiently. The meta-learner/second level learner is trained on the predictions obtained by integrating the capabilities of diverse algorithms or first-level learners known as individual learners.

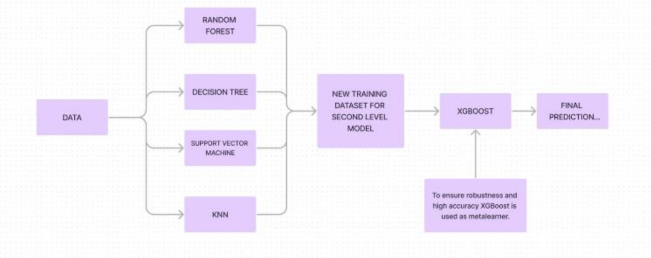

DATA
RANDOM FOREST
DECISION TREE
SUPPORT VECTOR MACHINE
KNN
NEW TRAINING DATASET FOR SECOND LEVEL MODEL
XGBOOST
FINAL PREDICTION...
To ensure robustness and high accuracy XGBoost is used as metalearner.

Figure 5: Diagrammatic representation of stack ensemble model with XGBoost meta-learner.

By combining the diverse base models, ensemble models harness their individual strengths and mitigate their weaknesses, leading to a robust and highly accurate predictive model. Following models have been used in the proposed solution:

1) *Extreme Gradient Boosting:* Its iterative boosting of weak learners, enhances the model's ability to handle complex patterns.

2) *Decision Trees:* Contribute through their simplicity and interpretability, as a base model.

3) *Logistic Regression:* Offers probabilistic predictions and insights into feature importance.

4) *Naïve Bayes:* Provides probabilistic classification using Bayes' theorem, assuming strong feature independence.

5) *Random Forest:* Aggregates the predictions from an ensemble of decision trees to improve generalization.

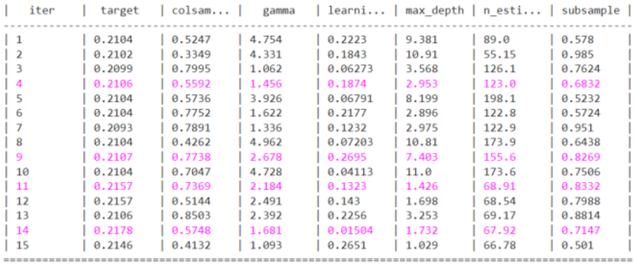
6) *Support Vector Machine (SVM):* Excels in finding hyperplanes that best separate classes in high-dimensional spaces.

7) *K-Nearest Neighbours (KNN):* Provides a non-parametric approach that captures local data structures.

8) *Convolutional Neural Network (CNN*): Leverages convolutional layers to capture hierarchical patterns and spatial feature.s

C. *Bayesian Optimization*

In order to optimize model performance, model parameters like learning rate, maximum depth of random forest etc., are properly tuned prior to model training. RandomSearch tuning often misses optimal solutions and GridSearch is computationally intensive while Bayesian Optimization efficiently finds the optimal hyperparameters [10], becoming the best choice for hyperparameter tuning. Figure 7 represents, *bayes_opt* iteratively finding optimal hyperparameters (highlighted ones) in a Random Forest Classifier.



```
|   iter    |  target   | colsam... |   gamma   | learni... | max_depth | n_esti... | subsample |
-------------------------------------------------------------------------------------------------
| 1         | 0.2104    | 0.5247    | 4.754     | 0.2223    | 9.381     | 89.0      | 0.578     |
| 2         | 0.2102    | 0.3349    | 4.331     | 0.1843    | 10.91     | 55.15     | 0.985     |
| 3         | 0.2099    | 0.7995    | 1.062     | 0.06273   | 3.568     | 126.1     | 0.7624    |
| 4         | 0.2106    | 0.5592    | 1.456     | 0.1874    | 2.953     | 123.0     | 0.6832    |
| 5         | 0.2104    | 0.5736    | 3.926     | 0.06791   | 8.199     | 198.1     | 0.5232    |
| 6         | 0.2104    | 0.7752    | 1.622     | 0.2177    | 2.896     | 122.8     | 0.5724    |
| 7         | 0.2093    | 0.7891    | 1.336     | 0.1232    | 2.975     | 122.9     | 0.951     |
| 8         | 0.2104    | 0.4262    | 4.962     | 0.07203   | 10.81     | 173.9     | 0.6438    |
| 9         | 0.2107    | 0.7738    | 2.678     | 0.2695    | 7.403     | 155.6     | 0.8269    |
| 10        | 0.2104    | 0.7047    | 4.728     | 0.04113   | 11.0      | 173.6     | 0.7506    |
| 11        | 0.2157    | 0.7369    | 2.184     | 0.1323    | 1.426     | 68.91     | 0.8332    |
| 12        | 0.2157    | 0.5144    | 2.491     | 0.143     | 1.698     | 68.54     | 0.7988    |
| 13        | 0.2106    | 0.8503    | 2.392     | 0.2256    | 3.253     | 69.17     | 0.8814    |
| 14        | 0.2178    | 0.5748    | 1.681     | 0.01504   | 1.732     | 67.92     | 0.7147    |
| 15        | 0.2146    | 0.4132    | 1.093     | 0.2651    | 1.029     | 66.78     | 0.501     |
=================================================================================================
```

Figure 7: Bayesian optimization iterations.

V. Results and Analysis

A. *Feature extracted Dataset*

Reduction to 1D feature space supports real-time applications and contributes to more efficient data analysis pipelines.

1) *Class Separation:* The transformation maximizes class separation by projecting data onto a single dimension, highlighting differences between classes.

2) *Enhanced Interpretability:* Simplifies complex data, making it easier to visualize and interpret class distinctions in the transformed feature space.

3) *Performance Evaluation:* Demonstrated notable performance improvements in subsequent classification tasks, with clear class boundaries facilitating higher accuracy.

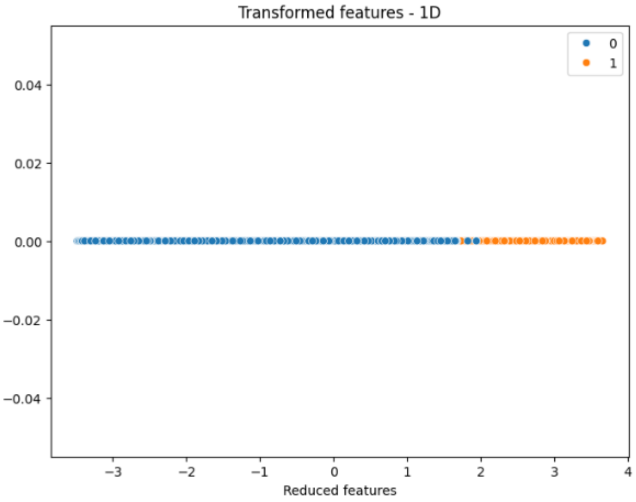


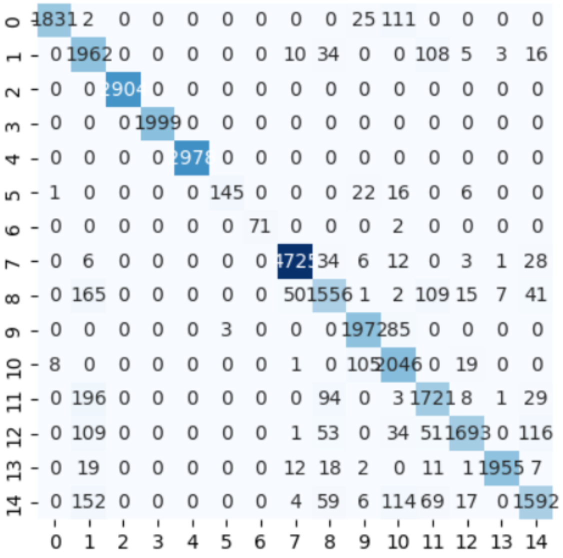
Figure 6: Feature space reduced and transformed to one-dimensional space.

B. *Binary Classification*

With soft voting, Max Voting Classifier combined the predictions from Naïve Bayes, Random Forest, XGBoost, and Logistic Regression base models and achieved an overall accuracy of 78% and a precision of 82%. The ensemble model leveraging the strengths of individual models, enhanced predictive performance through aggregated probabilities. Soft voting balanced bias and variance trade-offs yielding improved precision without significantly compromising accuracy.

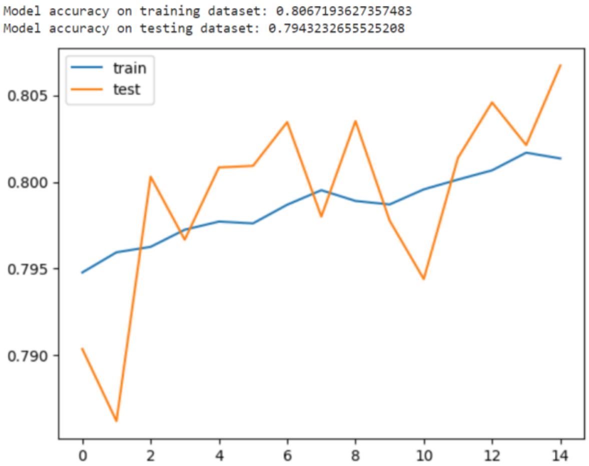
C. *Multiclass Clasification*

XGBoost meta model gives the final prediction leveraging the power of all the individual learners – Decision Trees, Random Forest, SVM, and KNN to give the best possible output, an accuracy of 93% with 92% precision. The diversity in base learners leads to better prediction. Stacking ensemble's accuracy outperformed CNN model's accuracy with early stopping callback in terms of multi-label classification which resulted in an accuracy of 81% with 80% precision.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1831	2	0	0	0	0	0	0	0	25	111	0	0	0	0
1	0	1962	0	0	0	0	0	10	34	0	0	108	5	3	16
2	0	0	2904	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1999	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	2978	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	145	0	0	0	22	16	0	6	0	0
6	0	0	0	0	0	0	71	0	0	0	2	0	0	0	0
7	0	6	0	0	0	0	0	4725	34	6	12	0	3	1	28
8	0	165	0	0	0	0	0	50	1556	1	2	109	15	7	41
9	0	0	0	0	0	3	0	0	0	1972	85	0	0	0	0
10	8	0	0	0	0	0	0	1	0	105	2046	0	19	0	0
11	0	196	0	0	0	0	0	0	94	0	3	1721	8	1	29
12	0	109	0	0	0	0	0	1	53	0	34	51	1693	0	116
13	0	19	0	0	0	0	0	12	18	2	0	11	1	1955	7
14	0	152	0	0	0	0	0	4	59	6	114	69	17	0	1592

Figure 7: Multi-label and multi-class classification confusion matrix



```
Model accuracy on training dataset: 0.8067193627357483
Model accuracy on testing dataset: 0.7943232655525208
```

Figure 8: CNN model's accuracy on training dataset vs testing dataset.

The proposed attestation framework utilizes advanced ensemble learning techniques to enhance the performance of both binary and multiclass classification tasks. These results underscore the effectiveness of ensemble approaches in leveraging the complementary strengths of various models, outperforming traditional methods and demonstrating their potential for complex classification problems.

VI. Conclusion and Future Scope

A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

1) For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

2) For papers with less than six authors: To change the default, adjust the template as follows.

a) Selection: Highlight all author and affiliation lines.

b) Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

[1] A. Niemi, Sampo Sovio, and J.-E. Ekberg, “Towards Interoperable Enclave Attestation: Learnings from Decades of Academic Work,” Apr. 2022, doi: https://doi.org/10.23919/fruct54823.2022.9770907.

[2] M. N. Aman, H. Basheer, J. W. Wong, J. Xu, H. W. Lim, and B. Sikdar, “Machine Learning Based Attestation for the Internet of Things Using Memory Traces,” *IEEE Internet of Things Journal*, pp. 1–1, 2022, doi: https://doi.org/10.1109/jiot.2022.3176530.

[3] G. Coker *et al.*, “Principles of remote attestation,” *International Journal of Information Security*, vol. 10, no. 2, pp. 63–81, Apr. 2011, doi: https://doi.org/10.1007/s10207-011-0124-7.

[4] Florian Kohnhäuser, Niklas Büscher, and S. Katzenbeisser, “A Practical Attestation Protocol for Autonomous Embedded Systems,” Jun. 2019, doi: https://doi.org/10.1109/eurosp.2019.00028.

[5] L. Moreau, E. Conchon, and D. Sauveron, “CRAFT: A Continuous Remote Attestation Framework for IoT,” IEEE Access, vol. 9, pp. 46430–46447, 2021, doi: https://doi.org/10.1109/access.2021.3067697.

[6] W. A. Johnson, S. Ghafoor, and S. Prowell, “A Taxonomy and Review of Remote Attestation Schemes in Embedded Systems,” IEEE Access, vol. 9, pp. 142390–142410, 2021, doi: https://doi.org/10.1109/access.2021.3119220.

[7] J. Lin and Q. Wu, “A Security Integrated Attestation Scheme for Embedded Devices,” 2021 7th IEEE International Conference on

[8] Sangapati Pavan, “Ensemble Learning Techniques,” Kaggle, 2020, doi: https://www.kaggle.com/code/pavansanagapati/ensemble-learning

[9] Rama Jayapermana,Aradea Dipalokareswara and Neng Ika Kurniati,"Implementation of Stacking Ensemble Classifier for Multi-class Classification of COVID-19 Vaccines Topics on Twitter",May 2022, https://www.researchgate.net/publication/363099952.

[10] Stamatios-Aggelos Alexandropoulos,Christos K. Aridas,Sotiris Kotsiantis, and Michael N. Vrahatis,"Stacking Strong Ensembles of Classifiers", May 2019, https://www.researchgate.net/publication/333109629.