I go through the survey paper named **A Survey on Machine Learning against Hardware Trojan Attacks: Recent Advances and Challenges**. This paper explores three types of HT circuits- 1) IP level 2)Bus level 3) SoC level. They also categorized protection against threat into four broad class solutions, namely1) HT detection, 2)DFS(design for security), 3) bus security, and 4) secure architecture. HT detection and DFS strategies are intended to defend against IP level HTs.

# Here I explore the ML technique both supervised and unsupervised.

# 1. As supervised learning require golden ICs, data with labels as a reference but unsuitable for large datasets

# 2. Unsupervised learning can overcome these deﬁciencies. However, unsupervised learning is sensitive to noise and readily falls into local optima. In particular, the output results for each training process cannot be determined clearly.

# Threat Model discusses the different scenario where HTs can inject like:-

# Untrusted foundry: An attacker at the foundry could implant an HT instance into the design by manipulating the lithographic masks. These HTs appear in the form of modiﬁcation to the transistors, gates, and interconnects into the original layout.

# Untrusted EDA tool, employee, or 3Pdesign house: Because increasingly more specialized IC designers and tools have become involved in the design-fabrication process, HTs might be inserted by Untrusted 3P commercial EDA tools or rogue designers within the in-house team (also called insider threats)[16]. Moreover, customers may also outsource their speciﬁcations to offshore 3P design houses, and these Untrusted design houses may add additional modules or functions to the original.

# Untrusted 3PIP vendor: SoC developers may purchase and employs 3PIP cores to complete their SoC designs. These IP cores provide by untrusted 3PIP vendors to customers may contain malicious logic or backdoors.

1. Untrusted routers or links in the on-chip bus: An adversary may compromise the integrity of the on-chip buses using malicious routing nodes or trafﬁclinks that have been infected with HTs. Such malicious bus fabrics would then be integrated into the SoCs.

# Untrusted SoC developer: This attack scenario assumes that an untrusted SoC developer may develop SoC designs infected with SoC-level HTs or integrate the soft/ﬁrm/hard IP cores from untrusted3PIP vendors containing such HT circuits that can separately or jointly impact other IP cores or the functions of the overall SoC functions

**HT Detection using ML-based techniques:**

HT detection techniques are typically used to verify whether any unwanted circuit is implanted in designed or fabricated ICs. Thus, ML-based techniques have seen progress in terms of (1) reverse engineering, (2) circuit feature analysis, and (3) side-channel analysis.

Madden et al. presented a spiking neural network (SNN)-based HT detection approach that was designed to detect DoS attacks through abnormal traffic patterns in NoC data. They ﬁrst extracted the spatial and temporal features of digital data exchanged across traffic links between adjacent routers. Then, these features were input into the SNNs and learned to reveal the abnormal operations. This method can effectively identify unseen attacks; however, the accuracy strongly depends on the length of the attacks.

K-means, ELM, DL, and BPNN are generally applied for HT detection; however, they have not yet been exploited for DFS, bus security, or secure architecture.

It is clear that ML algorithms are used mostly in **side-channel analysis**, probably because the side-channel information of ICs is easy to sample and vectorize. The second dominant feature is **the circuit feature analysis**. However, it is rarely used in reverse engineering because the process of feature extraction is complex and time-consuming, making it less practical.

We can conclude that SVM, including one-class and two-class SVM, is the most widely used supervised ML algorithm for identifying HT-infected ICs. However, SVM assumes that there are golden ICs available for training. Furthermore, K-means approaches are also relatively popular unsupervised learning algorithms for HT detection. However, they are not constrained by the above condition.

**HT THREATS BEYOND CHIPS**-

depending on the new threats, we categorize the HT threats

to the hardware ecosystem into four layers: the chip layer,

component layer, device layer, and behavior layer. Fig. 20

generalizes the four layers of Trojan threats in the hardware

ecosystem. Note that the chip-layer HT threats (the ﬁrst layer

of the four layers) include three types: IP-level HTs, bus-

level HTs and SoC-level HTs. These threats are introduced by

ﬁve typical scenarios during the IC design and manufacturing

process (see Section III.A), which represents the mainstream

of current HT research. In particular, the HT threats in each

the layer is described as follows:

-Chip Layer. IP-level, bus-level and SoC-level HT at-

tacks, i.e., chip HTs, are maliciously injected into chips

at any stage of the modern IC supply chain and can incur

IP-level, bus-level and SoC-level impacts on the chips

(see Table 6).

-Component Layer. Printed circuit board (PCB) designs

can be deliberately modiﬁed by tampering with the

interconnect lines at the internal layers or by altering

the components, thereby introducing PCB HTs (or com-

ponent HTs) [119], [120]. For example, an adversary

may inject such macro HTs into commercial-off-the-

shelf (COTS) modules or separate functional modules

or use PCB features, such as the joint test action group

(JTAG) interface and test hooks, to cause malfunctions

or secret information leakage [121], [123]. PCB HTs

can speciﬁcally affect the components themselves.

-Device Layer. Device HTs may exist in the functional

components and COTS, e.g., Bluetooth and WiFi mod

ules, of electronic devices and can affect the commu

nication and data transmission between each function

module of the electronic device. For example, an ex-

ternal memory module containing device HTs might

tamper with data during communication or maliciously

The new threat emerging in new layers depending on the new threats, we categorize the HT threats to the hardware ecosystem into four layers:**1) the chip layer 2) component layer,3)device layer and 4)behavior layer.**

**1) Chip Layer**: IP-level, bus-level and SoC-level HT attacks, i.e., chip HTs, are maliciously injected into chips at any stage of the modern IC supply chain and can incur IP-level, bus-level and SoC-level impacts on the chips.

2)**Component Layer**: Printed circuit board (PCB) design scan be deliberately modiﬁed by tampering with the interconnect lines at the internal layers or by altering the components, there by introducing PCB HTs (or components HTs) . For example, an adversary may inject such macro HTs into commercial-off-the-shelf (COTS) modules or separate functional modules or use PCB features, such as the joint test action group(JTAG) interface and test hooks, to cause malfunctions or secret information leakage. PCB HTs can speciﬁcally affect the components themselves

3)**Device Layer**: Device HTs may exist in the functional components and COTS, e.g., Bluetooth and WiFi modules, of electronic devices and can affect the communication and data transmission between each function module of the electronic device. For example, an external memory module containing device HTs might tamper with data during communication or maliciously intercept and deliver such data through the connected multiplexing interface

3)**Behavior Layer**: Abnormal behaviors and operations by electronic devices during runtime can be produced by extremely hidden HTs in hardware system or functional modules. In addition, behavior HTs in boot programs can mislead the device and make it load untrusted programs or change the ownership of an IoT device, thus affecting the identity authentication for device-to-device and device-to-user communication. For example, a malicious universal serial bus (USB) ﬂash disk (i.e., Bad USB) may disguise itself as another component, thereby affecting system functionality.

**I check many websites for ICs or network traffic datasets here I listed some of them**

* <https://ant.isi.edu/datasets/>
* <http://crawdad.org/>
* <https://networkdata.ics.uci.edu/resources.php>
* <https://archive.ics.uci.edu/ml/machine-learning-databases/cpu-performance/>
* <https://zenodo.org/record/375521#.XuIBZkUzbIU>

**Here I am going to provide some link of paper which I am gonna start reading to know about how to collect data and how to apply ML on HT detection.**

* SVM-based Real-Time Hardware Trojan Detection for Many-Core Platform
* Hardware Trojan Identification Using Machine Learning-based Classification(<https://pdfs.semanticscholar.org/e521/502a3580772c1cbac0f2fdc16a248f18c779.pdf>)

**The problem I am facing:**

Not able to understand how to collect data for performing ML on HTs. Provide me some help on this.