Reading HW2

> B11130225
>
> 資工四乙
>
> 鄭健廷

Core Issue

Zero-day malware, also known as zero-day threats, refers to malicious programs that have never been detected before or are too new for traditional antivirus software to recognize. Their novelty and lack of known defensive signatures make detection and defense exceptionally challenging. This paper primarily explores how to leverage deep learning, a powerful branch of machine learning, to address this severe cybersecurity threat.

Key Contributions and Content

Authors F. Deldar and M. Abadi systematically organize and analyze the application of existing deep learning techniques in the field of zero-day malware detection. Their primary contribution lies in proposing a taxonomy that categorizes relevant techniques into four major groups:

- **Unsupervised Learning:** These methods do not require large quantities of labeled malware samples. Instead, they focus on identifying anomalous or unusual patterns within data, making them suitable for detecting unknown threats.
- **Semi-supervised Learning:** Combines a small number of labeled samples with a large volume of unlabeled samples for model training. This approach holds practical value in cybersecurity, where obtaining large quantities of malware samples is often challenging.
- **Few-shot Learning:** Aims to enable models to learn to recognize new categories from a very small number of samples. This is crucial for rapidly responding to novel zero-day malware.
- **Adversarial Resistance:** Explores how to make deep learning models more resilient to “adversarial attacks.” Attackers may deliberately craft malicious samples designed to deceive models, making model robustness a vital research focus.

Analysis Dimensions of the Papers

Under these four main categories, the papers conduct detailed comparisons and analyses of various techniques, examining aspects including:

- **Deep Learning Architecture:** Examples include the type of neural network used (CNN, RNN, Autoencoder, etc.).
- **Feature Encoding:** How raw files or behaviors are converted into numerical features understandable by deep learning models.
- **Platform:** Primary operating systems (e.g., Windows, Android) or devices (e.g., IoT) targeted by the research.
- **Functionality:** Whether the model's primary task is limited to “detection” (determining if something is malicious) or extends to “classification” (identifying which malware family it belongs to).

QA

1. What is zero-day malware?

Zero-day malware is a brand-new, previously unknown malicious software.1 Because it's new, security vendors haven't had time ("zero days") to analyze it and create a specific signature or patch. This makes it invisible to traditional antivirus software that relies on databases of known threats, allowing it to be highly effective in its initial attacks.2

2. What are the different learning methods?

These are different strategies for training machine learning models, distinguished by the type of data they use.3

- **Unsupervised Learning:** The model is given a large amount of **unlabeled** data and must find patterns, structures, or anomalies on its own.4 It's like being given a pile of mixed-up photos and asked to sort them into groups without being told what the groups are.
- **Semi-supervised Learning:** This is a hybrid approach. The model is trained on a dataset that contains a **small amount of labeled data** and a **large amount of unlabeled data**.5 It uses the labeled data as a guide to help it make sense of the much larger unlabeled portion.6
- **Few-shot Learning:** The model is designed to learn and make accurate predictions for a new category after seeing only a **very small number of examples** (the "shots").7 This mimics how humans can often recognize a new type of object after seeing it just once or twice.8
- **Adversarial Learning (or Adversarial-Resistant Learning):** This isn't about learning *from* adversarial data, but about training a model to be **robust against it**. It involves intentionally creating tricky or deceptive inputs (adversarial examples) to try and fool the model, then using those examples to retrain the model to be stronger and less easily fooled. It's like a boxer sparring with a tricky opponent to learn how to defend against their specific moves.

3. How can these methods help defend against zero-day malware?

According to the paper's framework, each learning method addresses a specific challenge in detecting unknown threats:

- **Unsupervised Learning** is ideal for zero-day detection because it excels at **anomaly detection**.9 It can learn what "normal" file or network behavior looks like and then flag any new, unseen activity that deviates from that norm as potentially malicious, without needing a prior signature.10
- **Semi-supervised Learning** helps solve the problem of **data scarcity**.11 It's hard to get many labeled samples of a brand-new malware family. This method leverages the vast amount of available benign (harmless) files (unlabeled data) alongside a few known malicious samples (labeled data) to build a more accurate detector.
- **Few-shot Learning** is crucial for **rapid response**.12 When a new malware family emerges, analysts may only have a handful of samples. This method allows them to quickly train a model to recognize this new threat family from just those few examples, speeding up the defense process dramatically.
- **Adversarial-Resistant Learning** makes defenses **more resilient**.13 Malware authors actively try to modify their code just enough to evade detection models (an adversarial attack).14 By training models to be robust against such evasion techniques, we create more durable and reliable security systems.

4. Malware Detection (MD) vs. Malware Classification (MC)

The difference lies in the question they answer:

- **Malware Detection (MD):** This is a **binary** (yes/no) decision. Its only job is to determine if a file is malicious or benign (harmless).
  - *Question:* "Is this file dangerous?"
- **Malware Classification (MC):** This is a **multi-class** problem that happens *after* a file is detected as malware. Its job is to determine the specific *type* of malware it is.
  - *Question:* "Okay, it's dangerous. But *what kind* of danger is it—a Trojan, ransomware, or spyware?" 分類

In short, **MD finds the needle in the haystack**, and **MC figures out what kind of needle it is**.

5. Malware Category (C) vs. Family (F) Classification

This is about the level of granularity in Malware Classification:

- **Category (C):** This is a **broad** classification based on the malware's general behavior or purpose. Examples include **Ransomware**, **Spyware**, **Trojan**, or **Worm**.15 It tells you *what it does*.
- **Family (F):** This is a **specific** classification based on shared code, infrastructure, or origin. It groups malware variants that evolved from the same codebase. Examples include **WannaCry** (a Ransomware family), **Zeus** (a Trojan family), or **Emotet** (a Banking Trojan family). It tells you *who made it* or *its lineage*.

Think of it like classifying animals: **Category** is like saying "it's a dog," while **Family** is like saying "it's a Golden Retriever."

6. Few-shot Learning Terminology

In a few-shot learning task, the data is split in a specific way to simulate learning from few examples:

- **Support Set:** This is the small set of **labeled examples** the model gets to learn from. It acts as the "study material" or flashcards. For instance, you might give the model 3 examples of a new ransomware family and 3 examples of a new spyware family.
- **Query Set:** This is the set of **unlabeled examples** that the model must classify using the knowledge it just gained from the support set.16 This is the "test" to see if it learned correctly.

N-way K-shot describes the structure of the task:

- **N-way:** Refers to the number of different classes (**N**) in the support set.
- **K-shot:** Refers to the number of examples (**K**) for each class in the support set.17

Example: A **5-way 3-shot** task means the model is shown a support set containing **5** different malware families, with **3** examples from each family. It must then use that knowledge to classify new samples in the query set.

7. One-Shot Learning and Zero-Shot Learning

These are specialized, more challenging versions of few-shot learning:

- **One-Shot Learning:** This is simply an **N-way 1-shot** task. The model must learn to recognize a new class from seeing **only a single example**. It's like showing a child one picture of a zebra and expecting them to recognize all other zebras.
- **Zero-Shot Learning:** This is the most extreme case. The model must classify samples from classes it has **never seen any examples of** during training. It does this by learning a mapping between visual features and high-level semantic descriptions (attributes or text). For example, you would train a model on various known malware types and provide descriptions. Then, you give it a description of a new type, like "This malware steals cryptocurrency by hijacking the clipboard," and it must identify a file that exhibits this behavior, even without ever seeing a labeled example of a "clipboard crypto-hijacker" before.