Editorial: Adversaries and Robustness

Today's embedded computing is embracing AI-based techniques, such as decision making at the devices themselves using machine-learning-based models, especially in the *edge-computing* scenario, where locally made decisions help with low-latency reactivity and longer latency decisions are accomplished through communications with higher levels of the computing and network hierarchy. For example, in autonomous driving, not all decisions can be made centrally, as real-time reaction time is too short. Another example is edge-computing in a cloud-based monitoring and control in smart infrastructures.

In this context, adversaries could try to either pollute the machine-learning process through data corruption or pollute the models by using malware and other means. Also, even without compromising the learning or the model—adversarial inputs could be used through false data injection or misleading signals — to induce the system into making wrong decisions.

A couple of years ago, researchers showed examples in which models for visual recognition of traffic signs were tainted by adversaries who placed small stickers on traffic signs to retrain cars to ignore stop signs. They showed that this would make an autonomous car get into accidents. One could do the same for medical devices that use machine-learning models for administering insulin dosage to a patient. Similarly, models used for intrusion detection in a network firewall could be misled to allow malicious traffic into a secure network.

Several machine-learning models are used in cyber security these days. Examples include malware detection, web application firewall, and so on. It has been shown that with very small changes in the malware binary—which does not alter the malicious functionality of the malware—one could make a model misclassify a malware as a benign executable. A firewall can be misled to allow malicious traffic into the network. In such cases, neither the model nor the training gets compromised—but it is the input that is doctored to escape the model's discerning ability.

Such examples are plentiful, and every day more examples are reported in the literature. This gives us pause before we start making AI and machine learning ubiquitous in search of ever-increasing autonomy.

Fortunately, the researchers are not just hopeless witnesses of these adversarial effects, as they are working feverishly on robust machine learning against adversarial activities. Robustness in machine learning has always been an interesting area of research—but more to make the models robust against noise or outliers that are of benign origin, arising out of human errors or instrument or process anomalies. But when the enemy is cleverer, the robustness cannot just target such benign data corruption, it must target specific adversarial behavior.

For example, robust models that can resist the misleading image classification that will induce an accident by an autonomous car, cannot just depend on outlier detection-based data validation before applying the classification model. It must innovate robust training methods as well as secure the model from being retrained by an adversary before being deployed. A malware classifier that is misdirected by an adversarially crafted binary is made robust by training the models with synthetically created adversarial malware samples into the training set. The models deployed are cryptographically secured, so that any attempt to change the model will not match in hash values, and they are periodically checked for integrity. So, robust modeling, robust feature selection,

securing integrity of the models, and robust learning are becoming commonplace before serious deployment of machine-learning-based techniques in critical applications.

We do not see many submissions regarding robust machine learning, adversarial machine learning, or adversarial threat models as applied to embedded systems. This is certainly an area that deserves a special issue. I would invite anyone who is involved in this kind of research to come up with a special issue proposal on this topic. Even though the scope of this research is much wider than embedded computing, I believe there is plenty of scope within embedded computing to warrant a special issue.

The present issue (Vol. 18, No. 4) has a mix of papers, including thermally aware real-time scheduling, vehicular network architecture, QoS management in multi-core computing, hardware tracing hooks in multi-core, and so on.

We recently decided to start accepting “keynote” papers from established researchers who have a certain level of authority in a subfield of embedded computing, and “tutorial” papers from researchers who are willing to put together a tutorial on a subfield of embedded computing. We also invite short “industry experience” papers that industry researchers can submit on a very specific industry experience in building or validating a complex embedded system. These should ideally be such that there is a lesson to learn from the experience being described. Finally, if any author wants to get “artifact verified” notation on their article, we request them to contact us, so with their article they can submit “artifact” germane to the submission and we can get them verified.

Sandeep K. Shukla
Indian Institute of Technology Kanpur

Editor-in-Chief