Solution Brief

AI Technologies – Unleash AI Innovation in Network Applications

New instructions in the Intel® Xeon® Scalable processors combined with optimized software frameworks enable real-time AI within network workloads

Executive Summary

Silicon and software technology advancements by Intel targeting AI inferencing have lowered the barrier (compute cost and R&D effort) to unleash the creativity and innovation of the network application developers on the use of AI advanced techniques within their commercial solutions. Intel's new instructions targeting AI inferencing performance within the Intel® Xeon® Scalable processors in combination with the software enabling of industry standard frameworks and the network segment specific library (Traffic Analytics Development Kit) serves as a foundation to introduce real-time AI-based processing within networking workloads.

Given these new acceleration instructions are present with our latest Intel® Xeon® CPUs, AI innovation can be introduced in networking equipment from the data center out to the edge without the need of specialized hardware (for example, GPUs, Neural Processing Unit, and so on). This guide introduces the solution technologies followed by a detailed consumption and application benefits within targeted use cases.

This document is part of the Network Transformation Experience Kit, which is available at https://networkbuilders.intel.com/network-technologies/network-transformation-exp-kits.

Introduction

The current practice of using fixed patterns, signature matching, and rules to detect known patterns in network traffic is being replaced with artificial intelligence driven algorithms. Sophisticated malware, such as the recent SolarStorm attack, emphasizes the need for advanced detection methods to identify unknown types of malicious network traffic. Traditional methods fall short in detecting unknown types of malware-generated network traffic, which calls for more advanced detection techniques that incorporate inspection of the overall packet structure, rather than specific static patterns.

Industry practices are introducing AI techniques using machine learning and deep leaning models across network analysis approaches. Here is a small sampling of use cases:

- **Flow Analysis:** Used to identify anomalies within networks, to analyze encrypted network traffic, and to profile applications.
- **User intention-based traffic analysis**: Algorithms and frameworks to analyze user actions and network events on a host according to their credentials. This can lead to detecting relationships, identifying anomalies, and conducting empirical assessments of security.
- **Web page access classification:** Predict category of web page being accessed by user (adult content, gaming, news, and so on)
- **Traffic anomalies detection**: Detect flow outliers through use of statistical approaches, similarity approaches, and pattern mining approaches.
- **Malware detection:** Detect malicious content in portable execution files, JavaScript, or detection of Command and Control (C2) malware network attacks as described in Palo Alto Networks Unit 42 post: Using AI to Detect Malicious C2 Traffic.

Solution Description

The network industry is quickly adopting and innovating with the use of AI technologies across many solutions offered within traditional appliances and more recent cloud-based offering. For example, analysts are projecting the use of AI in Enterprise SD-WAN deployments to increase from 5% in 2021 to 40% in 2025[1].

Intel continues to lower the compute cost of adopting AI technologies by adding AI acceleration right in the Intel Xeon Scalable processors. The first AI acceleration technology by Intel, the Intel® Deep Learning Boost (Intel® DL Boost) featuring VNNI was first included in the 2nd Generation Intel® Xeon® Scalable processors. VNNI is a specialized instruction set that uses a single instruction for deep-learning computations that formerly required three separate instructions. The upcoming Sapphire Rapids CPU includes Intel® Advanced Matrix Extensions (Intel® AMX) instructions, a new expandable two-dimensional register file and new matrix multiply instructions to enhance performance for various deep learning workloads.[2]

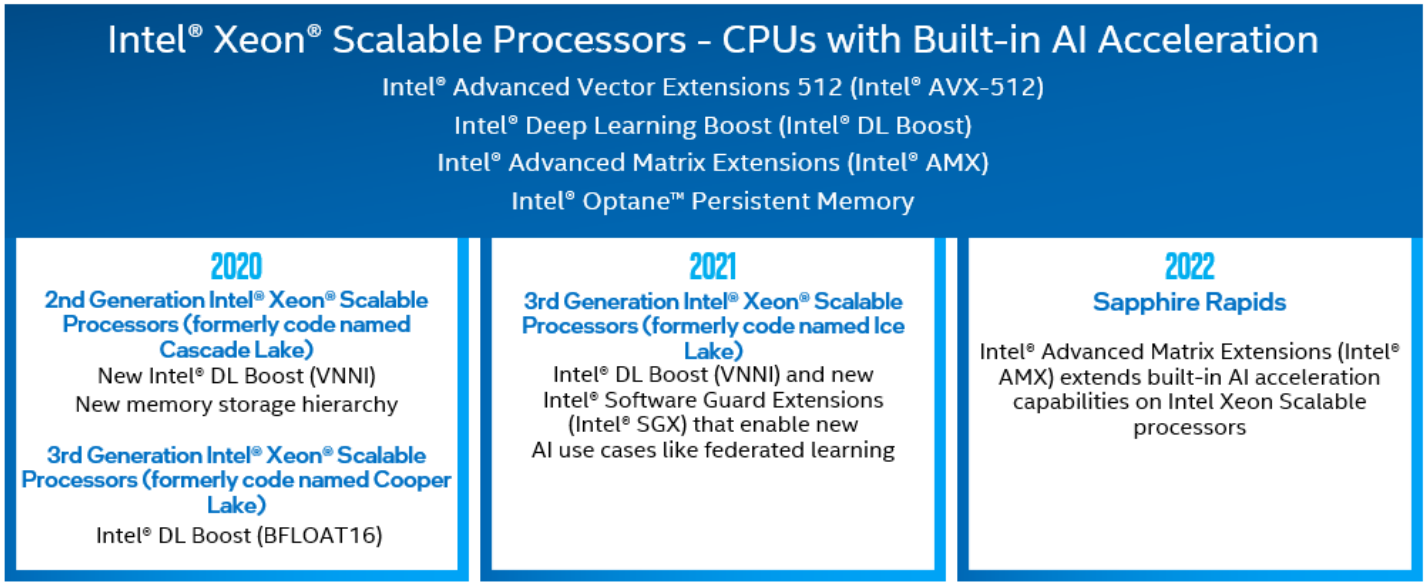

Figure 1 Intel® Xeon® Scalable Processors - CPUs with built-in AI acceleration

To take advantage of the VNNI instruction, Intel has updated the most prevalent AI industry frameworks including TensorFlow*, PyTorch*, MXNet*, PaddlePaddle*, and Caffe*.

In addition, Intel has packaged a set of tools and library components specific to networking application use. The Traffic Analytics Development Kit (TADK) is a starting point for the introduction of real-time artificial intelligence within network workloads including End-Point Security, Cloud Networking (SD-WAN), Next-Generation Firewalls, and Web Application Firewalls.

This solution guide first introduces the overall set of AI technologies available to the networking industry in innovative network security solutions and then reviews the consumption within the context of two specific workload examples: Web Application Firewall and Next Generation Firewall. We found that an AI driven approach reduced the compute cost some while also improving accuracy from less false-positives and less false-negatives in the Web Application Firewall. We also highlight how a TensorFlow, or TensorFlow Lite AI model is made to run most efficiently on Intel Xeon Scalable processors.

Technologies Implemented

The following diagram captures the key technologies supporting the introduction of AI advanced techniques within networking solutions, which include AI acceleration-specific CPU instructions, optimized AI industry frameworks and libraries, and developer tools and libraries.

[1] "SASE, AI Fuel SD-WAN Winners", SDXcentral article, September 27, 2021

[2] For workloads and configurations visit www.Intel.com/PerformanceIndex. Results may vary.

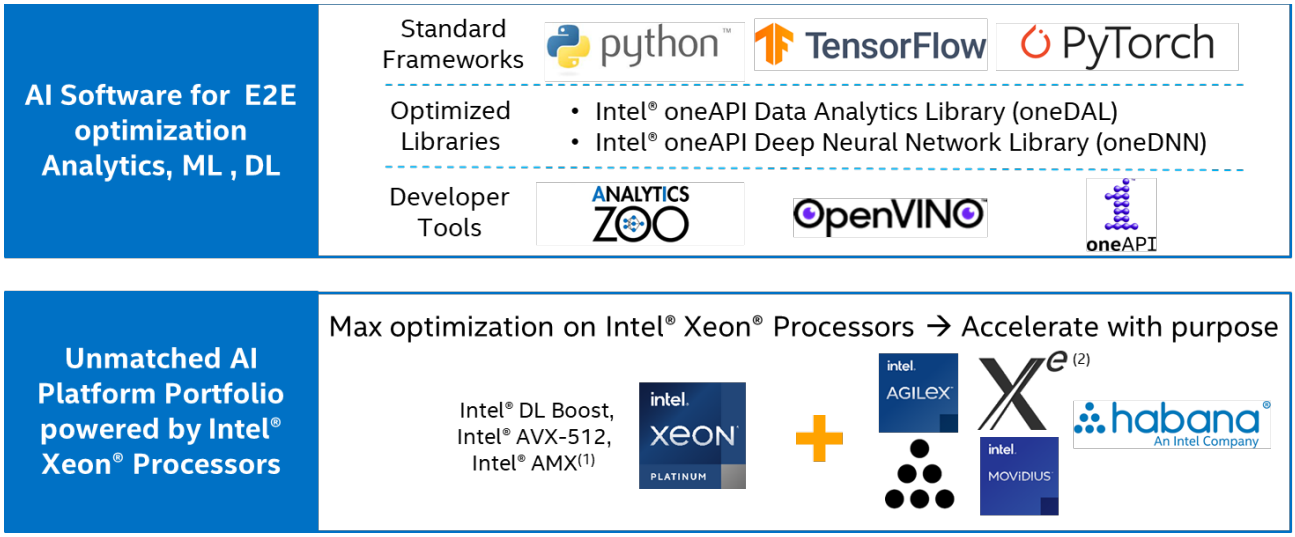

Figure 2 Key technologies that support AI advanced techniques

CPU Instructions

- **Intel® Deep Learning Boost:** A group of acceleration features introduced in the 2nd Generation Intel Xeon Scalable processors that aims to provide significant performance increases to inference applications built using leading deep-learning frameworks such as PyTorch, TensorFlow, MXNet, PaddlePaddle, and Caffe. The foundation of Intel Deep Learning boost is VNNI, a specialized instruction set that uses a single instruction for DL computations that formerly required three separate instructions.
- **Intel® Advanced Matrix Extensions (Intel® AMX) Instructions:** Hardware block in the Sapphire Rapids CPU with a new expandable two-dimensional register file and new matrix multiply instructions to enhance performance for various deep learning workloads.
- **Intel® Advanced Vector Extensions 512 (Intel® AVX-512):** A 512-bit instruction set that can help advance performance for demanding workloads and usages like AI inferencing.[3]

Optimized AI Industry Frameworks

- **Intel® oneAPI Data Analytics Library (oneDAL):** A library that helps speed up big data analysis by providing highly optimized algorithmic building blocks for all stages of data analytics (preprocessing, transformation, analysis, modeling, validation, and decision making) in batch, online, and distributed processing modes of computation.
- **Intel oneAPI Deep Neural Network Library (oneDNN):** Open-source cross-platform performance library of basic building blocks for deep-learning applications. The library is optimized for Intel® Architecture Processors, Intel® Processor Graphics and Xe Architecture graphics.
- **Intel® Neural Compressor:** Intel® Neural Compressor (formerly known as Intel® Low Precision Optimization Tool) is an open-source Python* library running on Intel® CPUs and GPUs, which delivers unified interfaces across multiple deep-learning frameworks for popular network compression technologies, such as quantization, pruning, and knowledge distillation. This tool supports automatic accuracy-driven tuning strategies to help users quickly find out the best quantized model. It also implements different weight pruning algorithms to generate a pruned model with predefined sparsity goals and supports knowledge distillation from the teacher model to the student model. The library is available across popular deep-learning frameworks such as TensorFlow, PyTorch, MXNet, and ONNX* (Open Neural Network Exchange) runtime.
- **Intel® Math Kernel Library (Intel® MKL):** This library has implementations of popular mathematical operations that have been optimized for Intel hardware in a way that lets applications take full advantage of the Intel instruction. It is compatible with a broad array of compilers, languages, operating systems, and linking and threading models.
- **Intel® Distribution for Python*:** Accelerates AI-related Python libraries such as NumPy, SciPy, and scikit-learn* with integrated Intel® Performance Libraries such as Intel MKL for enhanced AI inferencing.
- **Industry framework optimizations:** Intel has worked with Google* on TensorFlow, with Apache on MXNet, with Baidu* on PaddlePaddle, and on Caffe and PyTorch to enhance deep-learning performance using software optimizations for Intel Xeon Scalable processors in the data center, and it continues to add frameworks from other industry leaders.

[3] For workloads and configurations visit www.Intel.com/PerformanceIndex. Results may vary.

  - **Intel® Optimization for TensorFlow***: Binary distribution of TensorFlow with Intel® oneAPI Deep Neural Network Library primitives, a popular performance library for deep-learning applications. TensorFlow is a widely used machine-learning framework in the deep-learning arena, demanding efficient use of computational resources. To take full advantage of Intel® architecture and to extract maximum performance, the TensorFlow framework has been optimized using oneDNN primitives.
  - **XNNPACK backend for TensorFlow Lite:** XNNPACK provides highly optimized implementations of floating-point neural network operators built on SSE2, SSE4, Intel® AVX, Intel® AVX2, and Intel® AVX-512 instruction sets found on CPUs.

- **Hyperscan:** Intel created the Hyperscan open-source pattern matching project to enable an unprecedented level of performance and functionality in pattern matching. Hyperscan takes advantage of underlying CPU instructions (including Intel AVX-512) to deliver high speed pattern matching plays necessary in network and cloud security such as Web Application Firewall.

Networking Specific Developer Tools and Libraries

- **TADK:** A collection of optimized libraries and tools (see Figure 2) covering the needs of a typical end-to-end AI/ML pipeline used in Networking applications. TADK has a modular design supporting custom extensions and customer specific libraries to be included in the overall pipeline. TADK also includes sample open-source application integration (NGINX, FD.io VPP, ModSecurity) as well as sample trained models focusing on traffic classification and Web Application Firewall use cases.
  - **Flow Feature Extraction Library (FFEL):** Configurable and extendable library configured to obtain data and information about events that occur within a network flow, including:
    - **Packet feature**, including packet length, packet histogram, the sequence of lengths and arrival times of IP packets, up to some configurable number of packets.
    - **Protocol feature**, including the key fields of the packets data such as TLS cipher suites, SNI, DNS names/addresses, HTTP URI and header.
    - **Bag-of-words (BOW) feature**, token-encoding of the string fields in packets data.
  - **Lexical Parser:** Deterministic Finite Automaton (DFA) based tokenizer and token encoder.
    - DFA compiler takes a profile/dictionary to generate the run-time DFA engine.
    - Now supports SQL, HTML5, and Java Script tokenizer.
  - **Flow Classifiers:**
    - Bi-direction flow classification on 5-tuple and flow table management.
    - Time-wheel based flow aging mechanism.
  - **Protocol Detection:** protocol parser (libproto_proc.so) can detect and parse protocols including: IPv4, UDP, TCP, HTTP, TLS, QUIC, and DNS.
  - **AI Engine:** Intel oneAPI Data Analytics Library (oneDAL) is a powerful machine learning library that helps speed up big data analysis. TADK wrapped oneDAL library and uses its random forest algorithm to build the AI/ML classifier.
  - **DPI Engine:** the DPI engine can take the SNI field from the TLS traffic and identify the AppID by matching a ruleset based on Hyperscan.

Use Case Example #1 – Web Application Firewall

As a representative workload, we chose the popular open source ModSecurity Web Application Firewall (WAF) with support for OWASP core rule set - CRS. ModSecurity helps protect web applications by filtering and monitoring HTTP traffic between a web application and the Internet, reverse proxy.

We chose to narrow in on the components that detect the most common application attacks of SQL injection and XSS (Cross-Site-Scripting). ModSecurity detects these attacks within the Libinjection module using fingerprints and lexical analysis using traditional logic approach. Libinjection module is supported by two operators in the SecRule definition: detectSQLi (ModSecurity version 2.7.4 and later) and detectXSS (ModSecurity version 2.8.0 and later).

As highlighted in Figure 3, we replaced the Libinjection library with a trained machine-learning model. What we observed is that our AI driven approach reduced the compute cost some while also improving accuracy from less false-positives and less false-negatives. Our implementation and favorable results are due to the novel use of the Lexical Parser of the TADK framework where the combination of the DFA compiler creating a runtime DFA engine is responsible for the saving of compute cycles and overall inferencing time at the front end. The Lexical Parser DFA engine output then fed to a Random Forest Machine Learning model for inferencing, the model is trained by the sample data generated by sqlmap and XSStrike. The model is then executed in our AI engine taking advantage of the oneDAL Data Analytics Library to predict whether we have an attack or not[4].

[4] For workloads and configurations visit www.Intel.com/PerformanceIndex. Results may vary.

Please contact your local Intel sales representative for additional technical collateral and a copy of the modified ModSecurity software.

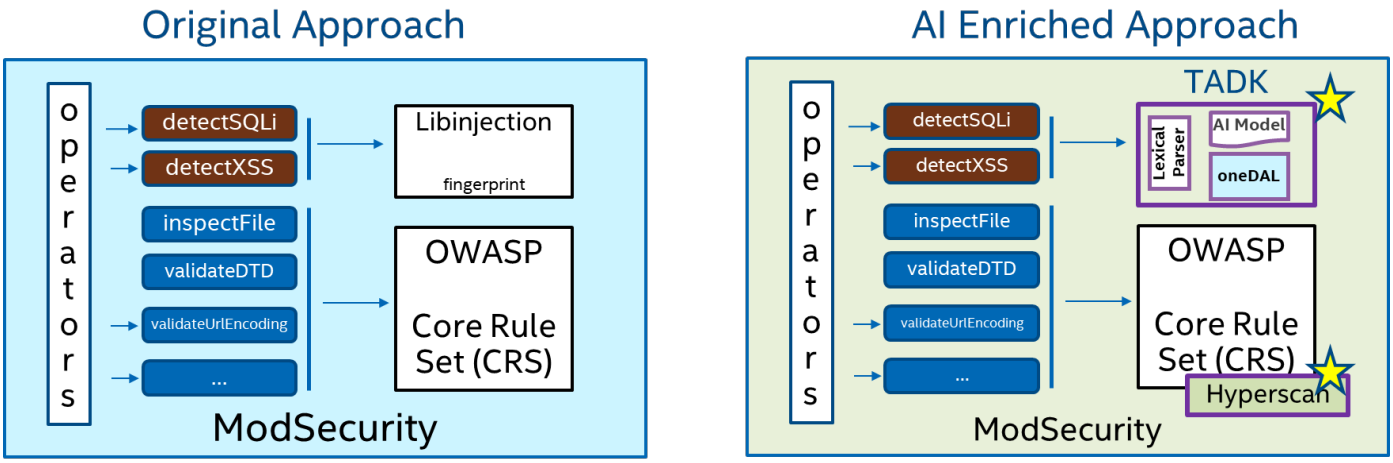

Figure 3 Introduction of AI Machine Learning within ModSecurity

Use Case Example #2 – Next Generation Firewall

A second use case to consider is the utilization of AI deep learning model to detect malicious JavaScript found in phishing kits, malvertising libraries or clickjacking campaigns where traditional signature or hash matching approaches are ineffective at detecting previously unknown malware.

Figure 4 is a representative implementation pipeline for the detection of malicious JavaScript with the AI technologies mapped onto the pipeline.

Similar to the Web Application Firewall, our TADK Lexical Parser is applied to the front-end of the pipeline to minimize the cost of parsing strings into engineered chars and tokens to feed a deep neural network trained model. We then utilized the Intel Neural Compressor (available with TensorFlow 2.6 or later) to convert the FP32 based model to an INT8 quantized model without loss of significant prediction accuracy. We also made sure that the TensorFlow base model would run most efficiently on Intel Xeon Scalable processors by including the oneDNN enabled version (available with TensorFlow 2.5 or later). Similarly, a TensorFlow Lite based model is made to run efficiently on Intel Xeon Scalable processors through the inclusion of the XNNPACK backend.

The DFA compiler creating a runtime DFA engine is responsible for saving compute cycles at the front end while the use of oneDNN minimized the compute cycles of the AI inferencing together satisfying our objective to add real-time detection within our allocated compute cost.[5]

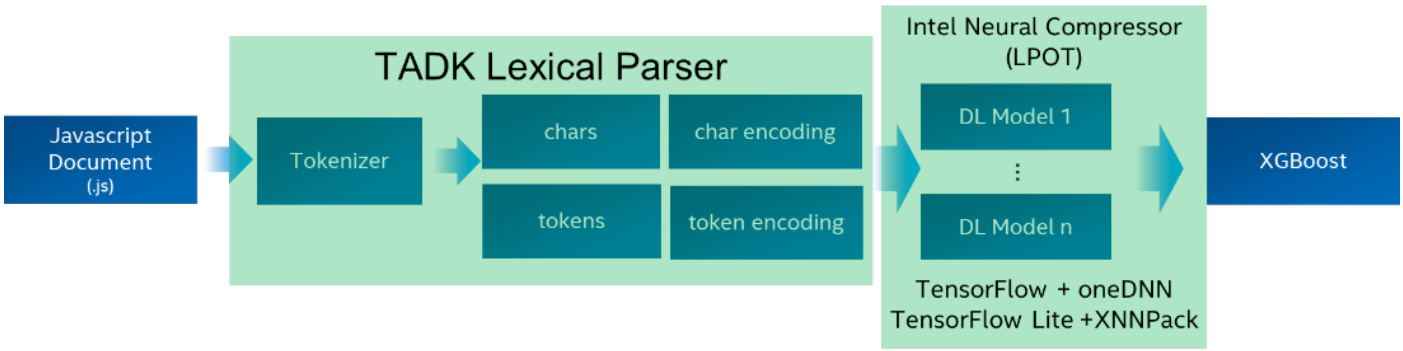

Figure 4 Pipeline to detect malicious JavaScript with the AI technologies

[5] For workloads and configurations visit www.Intel.com/PerformanceIndex. Results may vary.

Summary

Intel's continued investment in technologies aiding AI inferencing performance within the Intel Xeon Scalable processors combined with software enabling of industry standard frameworks and network segment specific library (Traffic Analytics Development Kit) are ready to unleash a whole new set of AI based innovation within networking workloads.

This solution guide sheds light on two such innovations identifying the use of key technologies and their role in realizing real-time AI inferencing within the overall cost of networking solutions based on Intel Xeon Scalable processors.

We welcome and encourage our partners to look at our two examples as a starting point on their journey to AI innovation or as a blueprint on how to minimize the cost of ongoing AI innovation. Incorporate these software libraries and frameworks and run them on the Intel Xeon platform that is already present in your networking solution as you introduce AI fueled innovation.

References

Table 1. References

TITLE	SOURCE
Maximize TensorFlow* Performance on CPU	Intel Developer Reference
Intel Network Transformation Kits	https://networkbuilders.intel.com/intel-technologies/network-transformation-exp-kits
Intel® oneAPI AI Analytics Toolkit	https://www.intel.com/content/www/us/en/developer/tools/oneapi/ai-analytics-toolkit.html
Intel oneAPI Deep Neural Network Library (oneDNN)	https://www.intel.com/content/www/us/en/developer/tools/oneapi/onednn.html
Intel® oneAPI Data Analytics Library (oneDAL)	https://www.intel.com/content/www/us/en/developer/tools/oneapi/onedal.html
Intel® Neural Compressor	https://www.intel.com/content/www/us/en/developer/tools/oneapi/neural-compressor.html
Intel® Math Kernel Library (Intel® MKL)	https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html
Intel® Distribution for Python*	https://www.intel.com/content/www/us/en/developer/tools/oneapi/distribution-for-python.html

1121/DN/WIPRO/PDF 684553-001US