


## Chapter 2

# Measuring Throughput and Latency of Machine Learning Techniques for Intrusion Detection

**Winfred Yaokumah**

 <https://orcid.org/0000-0001-7756-1832>

*University of Ghana, Ghana*

**Charity Y. M. Baidoo**

*University of Ghana, Ghana*

**Ebenezer Owusu**

*University of Ghana, Ghana*

### ABSTRACT

*When evaluating the effectiveness of machine learning algorithms for intrusion detection, it is insufficient to only focus on their performance metrics. One must also focus on the overhead metrics of the models. In this study, the performance accuracy, latency, and throughput of seven supervised machine learning algorithms and a proposed ensemble model were measured. The study performs a series of experiments using two recent datasets, and two filter-based feature selection methods were employed. The results show that, on average, the naive bayes achieved the lowest latency, highest throughput, and lowest accuracy on both datasets. The logistics regression had the maximum throughput. The proposed ensemble method recorded the highest latency for both feature selection methods. Overall, the Spearman feature selection technique increased throughput for almost all the models, whereas the Pearson feature selection approach maximized performance accuracies for both datasets.*

DOI: 10.4018/978-1-6684-9999-3.ch002

## INTRODUCTION

In recent times network security attacks and data breaches have become rampant while intrusion detection systems (IDS) are being developed with automated response mechanisms to detect and prevent such attacks (Anwar et al., 2017). An intrusion detection system provides active network security protection mechanisms against cybercriminal activities (Liang et al., 2019). These mechanisms monitor network operations and analyze packets for malicious activities (Paul et al., 2018). Packet analysis can take place on a network (network-based) or a computer (host-based). Network-based intrusion detection systems (NIDS) examine network behavior to determine if a node is under attack, whereas host-based intrusion detection systems (HIDS) check the logs kept by a single host for malicious operations (Chan et al., 2016). However, IDS sometimes fails to detect new external attacks and has a low accuracy rate, and a high false alarm rate (Khraisat et al., 2019; Liu & Lang, 2019). The inability of IDS to prevent network intrusion precisely and timely can jeopardize the information security goals of integrity, confidentiality, and system availability (Mishra & Yadav, 2020). In particular, the IDS that are implemented on networks can escalate the amount of delivery time, thereby reducing the reliability of network traffic (Xia et al., 2015). Also, packet processing may decrease network throughput and increase latency (Tsikoudis et al., 2016).

Two major overhead performance metrics, latency and throughput, are used to measure the IDS's timely detection of network intrusion. In the implementation of IDS, high throughput and low latency of an IDS are essential. Latency measures the time it takes for a processor to receive a request for a byte or message from memory (Hasan et al., 2021). In communication networks, latency or end-to-end delay represents the amount of time it takes for data to be received at its endpoint (destination). Intrusion detection systems frequently suffer from severe latency and network overhead, which make them irresponsive to attacks and the identification of malicious activities (Rahman et al., 2020). Likewise, throughput measures the speed at which the data is transmitted effectively (Ingley & Pawar, 2015). It describes the amount of data that is dispatched over a given time. According to Zhang et al. (2018), with high-speed traffic data, conventional intrusion detection systems experience latency and occasionally fail to identify intrusion patterns. However, to identify suspicious traffic momentarily, a real-time network IDS with machine learning (ML) techniques can promptly process vast volumes of network traffic data.

Specifically, machine learning (ML) techniques are enabling the modeling of IDS to provide significantly higher rates of intrusion detection (Srivastava et al., 2019). Some intrusion detection systems use ML classification algorithms to categorize network traffic as normal or irregular (Kaya, 2020; Thaseen & Kumar, 2017). For instance, the IDS implementation that employs ML approaches for classification includes Naive Bayes (NB), Adaptive Boost, PART (Kumar & Doegar, 2018), Active learning Support Vector Machine, Fuzzy C-Means clustering (Kumari & Varma, 2017), Multi-Layer Perceptron, Bayesian Network, Support Vector Machine (SVM), Adaboost, Random Forest (RF), Bootstrap Aggregation, Decision Tree (Halibas et al., 2018), Random Forest (Park et al., 2018); SVM (Kotpalliwar & Wajgi, 2015), Genetic Algorithm, and Support Vector Machine (Gharaee & Hosseinvand, 2017). These ML algorithms may provide varying levels of accuracy, latency, and throughput.

According to Yu et al. (2018), measuring the overhead performance of the learning algorithms accurately is important for determining their effectiveness. However, currently, few studies investigate the efficiency of IDS, considering its training time, testing time, latency, throughput, and detection time (Maseer et al., 2021). Thus, this experimental study focuses on supervised machine learning algorithms to measure latency and throughput using two intrusion detection datasets and two filter-based feature

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/measuring-throughput-and-latency-of-machine-learning-techniques-for-intrusion-detection/334467](http://www.igi-global.com/chapter/measuring-throughput-and-latency-of-machine-learning-techniques-for-intrusion-detection/334467)

## Related Content

---

### Generating an Artificial Nest Building Pufferfish in a Cellular Automaton Through Behavior Decomposition

Thomas E. Portegys (2019). *International Journal of Artificial Intelligence and Machine Learning* (pp. 1-12). [www.irma-international.org/article/generating-an-artificial-nest-building-pufferfish-in-a-cellular-automaton-through-behavior-decomposition/233887](http://www.irma-international.org/article/generating-an-artificial-nest-building-pufferfish-in-a-cellular-automaton-through-behavior-decomposition/233887)

### Research Information: Bio-Inspiration in Information Retrieval

Bouras Youcef (2021). *Advanced Deep Learning Applications in Big Data Analytics* (pp. 218-272). [www.irma-international.org/chapter/research-information/264556](http://www.irma-international.org/chapter/research-information/264556)

### MHLM Majority Voting Based Hybrid Learning Model for Multi-Document Summarization

Suneetha S. and Venugopal Reddy A. (2019). *International Journal of Artificial Intelligence and Machine Learning* (pp. 67-81). [www.irma-international.org/article/mhlm-majority-voting-based-hybrid-learning-model-for-multi-document-summarization/233890](http://www.irma-international.org/article/mhlm-majority-voting-based-hybrid-learning-model-for-multi-document-summarization/233890)

### A Fuzzy-Based Intelligent Picking and Placing Warehouse Robot

Himanshu Bothra, Shreyas Jha, Kumar Kanishka Dey, Anish Pandey and Ashwani Kumar (2022). *Empirical Research for Futuristic E-Commerce Systems: Foundations and Applications* (pp. 71-96). [www.irma-international.org/chapter/a-fuzzy-based-intelligent-picking-and-placing-warehouse-robot/309669](http://www.irma-international.org/chapter/a-fuzzy-based-intelligent-picking-and-placing-warehouse-robot/309669)

### Task Offloading Using Deep Reinforcement Learning for Edge IoT Networks

Pradeep Bedi, S. B. Goyal and Jugnesh Kumar (2022). *Demystifying Federated Learning for Blockchain and Industrial Internet of Things* (pp. 43-57). [www.irma-international.org/chapter/task-offloading-using-deep-reinforcement-learning-for-edge-iot-networks/308112](http://www.irma-international.org/chapter/task-offloading-using-deep-reinforcement-learning-for-edge-iot-networks/308112)