Information Security Journal: A Global Perspective

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uiss20

Assessing IoT intrusion detection computational costs when using a convolutional neural network

Mathew Nicho, Brian Cusack, Christopher D McDermott & Shini Girija

To cite this article: Mathew Nicho, Brian Cusack, Christopher D McDermott & Shini Girija (24 Apr 2025): Assessing IoT intrusion detection computational costs when using a convolutional neural network, Information Security Journal: A Global Perspective, DOI: 10.1080/19393555.2025.2496327

To link to this article: https://doi.org/10.1080/19393555.2025.2496327

Published online: 24 Apr 2025.

INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE
https://doi.org/10.1080/19393555.2025.2496327

Taylor & Francis
Taylor & Francis Group

Assessing IoT intrusion detection computational costs when using a convolutional neural network

Mathew Nicho[a], Brian Cusack[b], Christopher D McDermott[c], and Shini Girija[d]

[a]Research and Innovation Center, Rabdan Academy, Abu Dhabi, UAE; [b]Cloud Security Research Center, AUT University, Auckland, New Zealand; [c]School of Computing Science and Digital Media, Robert Gordon University, Aberdeen, UK; [d]College of Technology Innovation, Zayed University, Dubai, United Arab Emirates

ABSTRACT

IoT systems face vulnerabilities due to their data processing requirements and resource constraints. With 13 billion connected devices globally, this research investigates the economic viability of AI-based intrusion detection systems (IDSs), specifically analyzing the automation costs of implementing a Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) for classifying malicious sensor traffic. This study introduces an innovative framework that evaluates six distinct architectural components of CNN and LSTM: image input processing, convolutional layer operations, max pooling layer functionality, fully connected layer characteristics, softmax output activation, and class determination mechanisms. The framework employs six metrics: matrix size, feature vector number, input vector size, output vector size, and number of runs for dual data points. Experiments on the IoT-23 dataset showed our proposed CNN model outperformed LSTM, achieving 93% accuracy for binary classification and 96% for multi-class classification. The trained CNN demonstrated predictable resource utilization with increasing classification complexity, providing a framework for quantifying IoT IDS costs. The proposed framework provides a systematic methodology for evaluating machine learning classifiers in IoT environments, using quantitative metrics to assess implementation and operational costs, enabling data-driven selection of optimal security solutions based on specific deployment constraints.

KEYWORDS

Artificial intelligence; convoluted neural network; cost evaluation; internet of things; network security

1. Introduction

The widespread deployment of resource-constrained IoT devices across various economic sectors necessitates the development of lightweight, high-performance intrusion detection models (Wang et al., 2024). In this regard, organizations consistently evaluate cost economies for both direct and indirect cybersecurity implementation initiatives (Rodrigues et al., 2019). Hence, the application of cost-benefit analysis for security system selection and management has become critical as security breaches increasingly result in monetary damage, litigation costs, and loss of credibility (Cavusoglu et al., 2004). Moreover, the proliferation of smart devices has expanded the attack surface, creating more opportunities for system breaches and unauthorized control of IoT devices (Alam & Khan, 2022).

Security and privacy are major concerns for IoT devices worldwide (Mohanta et al., 2020), particularly critical across all IoT applications (Hassija et al., 2019). The exponential growth of IoT in commercial and individual spheres has led to increased successful cyber-attacks (Kuzlu et al., 2021). Furthermore, IoT faces significant challenges in managing traffic loads and models due to the rapid surge in internet-connected devices generating unprecedented data volumes (Čolaković & Hadžialić, 2018). Hence, security personnel must balance the management of extensive logs and events against implementation costs. While previous intrusion detection research has prioritized accuracy, the computational efficiency of machine learning (ML) classifiers remains understudied (Baig et al., 2021).

1.1. Optimizing IoT infrastructure: balancing data volume with computing costs

The IoT refers to sensor-based physical devices with processing power that communicate data

CONTACT Mathew Nicho mnicho@ra.ac.ae Research and Innovation Center, Rabdan Academy, PO Box 114646, Abu Dhabi, UAE

over networks for intelligent purposes, such as home or vehicle automation (Neshenko et al., 2019) with limited computational power and heterogeneous properties (Mohanta et al., 2020). To address IoT security concerns, machine learning (ML) and deep learning (DL) techniques can be leveraged to handle the heterogeneity and resource constraints of IoT network nodes, manage the massive real-time data generated by IoT devices, and adapt to the networks' extensively dynamic behavior (Hussain et al., 2020). However, research on the processing costs of employing ML methods for IoT intrusion detection is scarce. Therefore, this research represents a pioneering effort in the field of IoT intrusion detection by investigating the cost dynamics associated with using ML. The five-layer IoT design adds perception, transport, processing, application, and business layers to the basic foundation of sensing (layer 1), processing (layer 2), and applications (layer 3) that the three-layer architecture offers (Baziyad et al., 2022). In this expanded architecture, the perception (physical) and application layers serve functions corresponding to those in the three-layer architecture (Al-Awami et al., 2023).

IoT devices operate in an information-rich environment with intelligent human-machine interactions, making them vulnerable to diverse cyberattacks that demand innovative information security solutions (Sengupta et al., 2020). While the IoT literature has produced multiple reference architectures to provide an overview of various applications or focus on specific implementations (Ghirardello et al., 2018), there is no universal consensus on the most suitable IoT architecture (Sethi & Sarangi, 2017a, 2017b). This lack of standardization creates challenges for the information security sector in defining IoT entities, specifying their vulnerabilities, and developing standardized security measures that effectively balance processing effort, time, and costs across the ecosystem.

The IoT is a computing work system used for critical information processing and control actualization in automated and semi-automated environments (Serror et al., 2020). While IoT has improved company productivity and enhanced quality of life through automation, its widespread implementation across business and personal domains has significantly expanded the attack surface available to hackers (Lee & Lee, 2015), increasing the effort and cost required for network and systems security. The relationship between information security and user costs follows a security cost function, where higher security levels require greater user effort, directly impacting the user-friendliness of IoT devices and ecosystems (Shetty et al., 2010). Given the convenience and exponential growth of IoT devices, combined with the lack of architectural consensus and their broad application across various ecosystems (including personal area networks, smart homes, LANs, and SCADA networks), evaluating the cost-benefit tradeoff of IoT security implementation has become critical.

1.2. ML for IoT sensor security: addressing attack attribution challenges

Research has shown that sensors represent the weakest link in the IoT information chain, from data collection to application layer and servers, due to their large data volumes, limited computing power, and inadequate security features (Alladi et al., 2020). Figure 1 illustrates these vulnerabilities and maps the security problem area from a sensor classification perspective.

Recent research proposes addressing IoT security vulnerabilities through deep learning (DL) models and autonomous intrusion detection systems (IDS) (Chaabouni et al., 2019; Cui et al., 2018; Hemalatha et al., 2021; Liu et al., 2017; Tahsien et al., 2020; Zoppi et al., 2021). While these tools effectively classify large volumes of sensor data, they struggle to construct comprehensive feature sets needed for training autonomous agents. Mijalkovic and Spognardi (Mijalkovic & Spognardi, 2022) identified performance issues in current DL and automated IoT IDSs, highlighting inadequate accuracy rates for critical security threats like spoofing and data leakage attacks, and recommended further research to reduce false negative rates.

Although DL and conventional ML models have improved intrusion detection accuracy, their implementation demands significant effort, processing power, and cost. This challenge is amplified by the expanding attack surface, making intrusion detection scaling more complex since each ML

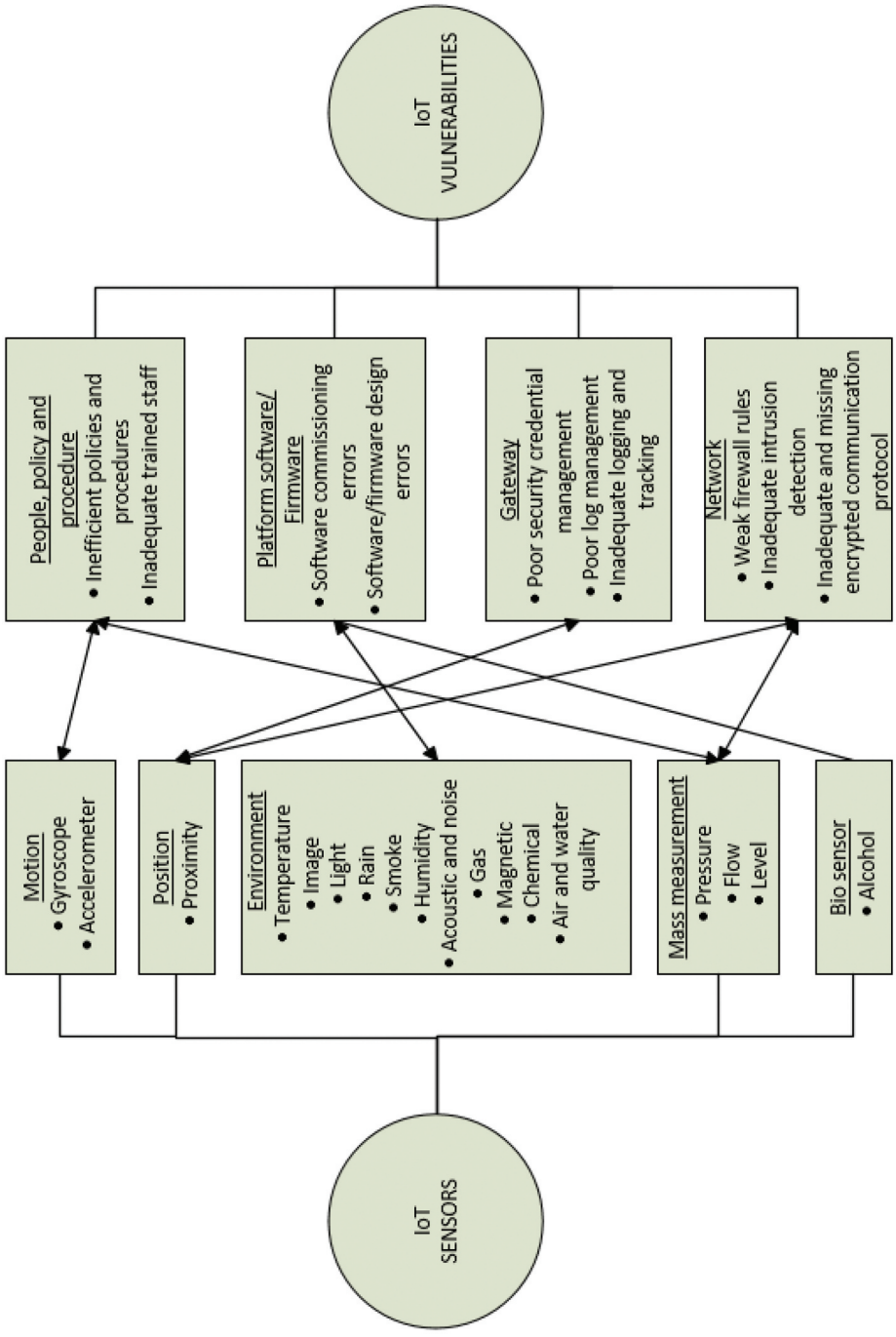

Figure 1. IoT sensor vulnerabilities note. Adapted from Nicho and Girija (2021).

model has different capabilities and costs. Additionally, assigning an optimal mix of cost-effective classifiers for each data sample presents significant challenges (Birman et al., 2022). Various approaches to IoT traffic packet classification include probabilistic, regression, and computational methods, with recent research emphasizing computational methods to achieve autonomous intrusion detection (Gibert et al., 2019). Studies have demonstrated successful binary classification results, with some extending to multiclass classification (Gibert et al., 2020). Specifically, network security requires a costing of protection against the potential benefits of any security proposal (Gordon & Loeb, 2006).

Considering the increased deployment of IoT sensors and their ecosystem communications, there is a critical need to optimize intrusion detection while minimizing associated costs. Despite the growing adoption of machine learning (ML) for IoT security, significant research gaps exist in: quantifying detection effort and resource requirements; understanding cost-performance tradeoffs across ML algorithms; and analyzing real-time processing needs in heterogeneous networks. The lack of standardized metrics for evaluating cost economies in ML-based detection, combined with insufficient research on scalability impacts, creates substantial challenges, particularly in multiclass computational classification for IoT intrusion detection. Therefore, this research aims to develop and validate quantitative metrics to assess the efficiency and cost-effectiveness of ML-based multiclass intrusion detection in IoT environments, with specific focus on network security implementation costs.

Multiclass classification enhances intrusion detection by enabling sequential analysis, first separating malicious from benign packets, then performing detailed classification of either category. This approach provides dual benefits: benign traffic classification reduces false positives while improving feature selection, and malicious traffic classification reveals attack patterns, sources, and characteristics. The resulting traffic profiles also generate performance metrics crucial for predicting computational risks and optimizing algorithm training.

1.3. Research contribution

Given the limited research on the economic viability of ML approaches in IoT intrusion detection, this study makes the following key contributions:

(1) The research introduces a pioneering quantitative framework using six metrics (matrix size, feature vector number, input vector size, output vector size, and dual data point runs) to evaluate neural network architectures. Our analysis of CNN architectural components (input processing, convolutional operations, max pooling, fully connected layers, softmax activation, and class determination) provides both a systematic industry approach and a replicable research methodology for evaluating ML classifiers.
(2) It provides a standardized framework to assess implementation costs against benefits for IoT intrusion detection systems, evaluating both binary and multiclass classification capabilities to determine the economic efficiency of automated IoT IDSs.
(3) The cost economy model provides a comprehensive performance and efficiency comparisons between CNN and LSTM models, examining classification metrics, resource utilization, and cost-effectiveness in binary and multiclass classification tasks to guide ML architecture selection.
(4) The research provides evidence-based guidance for IoT security investments through practical cost estimation models and implementation guidelines, helping decision-makers optimize protection levels based on organizational risk tolerance and resources.

The rest of the paper is structured as follows. After introducing the research topic, the introduction section identifies the research problem, gap, novelty, and contribution in Section 1. Section 2 reviews the literature on ML applications in intrusion detection, exploring the cost – benefit tradeoff of ML classifiers. Section 3 outlines the research methodology, Section 4 presents the results, followed by a discussion of the findings in Section 5, and Section 6 provides the conclusion along with limitations and areas for future research.

2. Literature review

IoT intrusion detection research addresses the complex challenge of differentiating between malicious and benign network traffic in real-time. A fundamental challenge in this domain is processing, classifying, and responding to massive data volumes efficiently. Within IoT ecosystems, data volume is determined by multiple factors: the quantity of connected sensors, data transmission frequencies, and stimulus management parameters (Alladi et al., 2020). A significant research gap exists in the systematic identification and analysis of variables affecting computational costs in IoT intrusion detection systems. This gap, combined with the challenges of data volume management, has driven research toward automated classification systems and autonomous security risk management solutions. The following subsections examine the current literature regarding IoT security vulnerabilities, IDS automation approaches, and security investment decision frameworks

2.1. Application of machine learning for IDS in IoT: the rising trend and challenges

Since the turn of the twenty-first century, sensor-based IoT devices have experienced exponential growth across industry sectors and private homes (Schiller et al., 2022). However, insufficient security vetting of IoT devices has led to a proliferation of IoT-related vulnerabilities (Najar-Pacheco et al., 2019), making IDSs essential for preventing, detecting, and mitigating threats that exploit security weaknesses in IoT devices and their broader ecosystem (Elrawy et al., 2018). Therefore, designing an IDS that effectively detects intrusions while maintaining IoT network scalability, reliability, and defense against targeted threats poses significant challenges (Asharf et al., 2020).

Machine learning classifiers have emerged as an effective solution (Thamilarasu & Chawla, 2019) for intrusion detection in computer systems (Musleh et al., 2023), requiring fewer computational resources for data processing and classification (Baraneetharan, 2020). Deep learning methods can process massive datasets with minimal preprocessing time (Ashiku & Dagli, 2021), though they demand substantial memory and computing power (H. Zhang et al., 2018). Recent experimental results using Tree-CNN classifiers, which integrate hierarchical tree structures into neural network architecture, have demonstrated reduced complexity and resource requirements (Mendonça et al., 2021). CNNs offer rapid training due to their architecture (Kim & Aminanto, 2017) and have shown exceptional performance across diverse applications including image classification, pattern recognition, and multimedia compression, while reducing parameters and improving processing speed in both training and inference phases (Q. Zhang et al., 2019). Consequently, evaluating appropriate classifiers for IoT-based IDS automation while considering cost-benefit factors can significantly advance detection techniques and IDS applications within deep learning.

The ability of LSTM models to efficiently process and analyze sequential data from connected devices makes them essential for IoT systems (Yu et al., 2021). LSTMs excel at handling time-series data from IoT sensors by preserving long-term relationships in sequential patterns. Their capacity to record temporal patterns makes them particularly effective for anomaly detection, network traffic forecasting, and predictive maintenance in IoT environments (Liu et al., 2019). We selected LSTM for comparison due to its superior performance in detecting sophisticated attacks like Advanced Persistent Threats (APT), compared to alternative approaches such as GRU (Dey et al., 2021; Eke et al., 2019).

2.2. Automating intrusion detection systems

The automation of intrusion detection systems requires real-time classification algorithms and intelligent processing of large-scale data for critical decision-making (Alhowaide et al., 2021). These automated IDSs employ classification algorithms that categorize IoT traffic as either malicious or benign. While binary classifiers provide basic system protection, more complex multi-class classifiers enable detailed traffic analysis for enhanced intelligence gathering (Hemalatha et al., 2021). Even when binary classification successfully protects the system, analyzing rejected traffic provides valuable insights for predicting future attack patterns.

Numerous researchers have pursued automated IoT intrusion detection solutions (Chaabouni et al., 2019; Cui et al., 2018; Tahsien et al., 2020; C. Zhang et al., 2022; Zoppi et al., 2021). Current research focuses on developing unsupervised deep-learning systems capable of continuously learning IoT traffic features while autonomously managing large packet volume (C. Zhang et al., 2022). While these solutions demonstrate success under controlled conditions with specific training datasets, comprehensive general-purpose solutions remain under development.

Deep learning solutions for IoT intrusion detection have been investigated through different ML algorithms (Liu et al., 2017; Ye et al., 2018). Artificial neural networks (ANNs) are versatile machine learning models that can be used for both supervised and unsupervised learning tasks, including pattern recognition and classification problems. While statistical clustering is an unsupervised ML technique that relies on data associations and characteristics, multiple regression is a supervised learning approach. Clustering identifies natural groupings in data, whereas ANNs rely on representing patterns in networks inspired by biological neural networks (Dong et al., 2021).

ANNs are designed so that each artificial neuron connects to many other neurons, and multiple neurons work together to process information. The output layer is where an ANN produces its final output, which could be a classification decision, regression value, or other solution depending on the task. However, the limitation of ANNs' application in IoT intrusion detection implementation is cost (Tahsien et al., 2020), as ANNs require large feature sets for training, which are limited by the resource budget for capturing the feature set scope. ANNs' structure also requires direct neuron connections that become more resource expensive as the size of an ANN increases to satisfy increasing problem complexity. IoT intrusion detection computation costs escalate with loading and can potentially overload an ANN. The computational cost of an ANN's multi-layered connected structures is reduced when the number of neural connections is reduced (C. Zhang et al., 2022). Hence, CNNs are a specialized class of ANNs that reduce the number of neural network connections required for IoT intrusion detection through parameter sharing and are more computationally efficient than fully-connected ANNs.

CNNs retain the ANNs' input, hidden, and output layers structure while creating computational cost efficiencies by reducing the number of connections between neurons (Zoppi et al., 2021). For example, a $16 \times 16\times$ 1-pixel input to a CNN has a 250% computational cost reduction advantage over an ANN with the same input in the first layer. In an ANN, each neuron is directly connected. Hence, each image input would have $16 \times 16\times 1$ pixels, based on grayscale weightings from 0 to 255 plus 1 for the bias, which equals 257 connections per neuron. However, the CNN is more cost-effective, and each convolution layer forward feeds $3 \times 3\times 1$ weightings plus 1 for the bias, for a total of 10 connections, hence allowing significant cost savings in each layer for computation. Each image data array is fed into the convolution function to reduce the amount of data going to the neurons in the fully connected layers. Normalization and pooling further reduce the data size, and the rectified linear unit (ReLU) and SoftMax functions are selected as cost-effective choices in CNN design (Ni et al., 2018). Hence, CNNs are supported in IoT IDSs research as a cost-effective solution to the challenges posed by IoT big data.

Recurrent neural networks (RNNs) are a type of neural network that process data sequentially by retaining information from earlier inputs through feedback loops (Sherstinsky, 2020). However, problems like the vanishing gradient problem make it difficult for ordinary RNNs to maintain long-term dependencies. This limitation is addressed by LSTMs, a specific type of RNN that can successfully recognize and maintain long-range patterns in sequence data by incorporating memory cells (Dey et al., 2021; Eke et al., 2019). Hence, LSTMs are widely used in IoT IDS research as a cost-effective solution to the challenges posed by IoT big data.

Recent studies have focused on employing DL methods such as CNN to create effective IDS for IoT scenarios. Hairab et al. (2022) proposed a CNN-based approach with regularization techniques to identify zero-day attacks in IoT networks (Chen et al., 2025). introduced a novel and impactful synaptic CNN to effectively detect intrusions

within dynamic IoT environments. Another hybrid CNN-LSTM architecture achieved high accuracy for both binary (93%) and multi-class (92%) classification on the UNSW-NB15 dataset, with even better performance on the X-IIoTID dataset (Altunay & Albayrak, 2023). Salih and Ibrahim (2023) demonstrated the application of deep learning in IoT forensics, emphasizing the superior performance of LSTM and RNN in effectively classifying IoT data for improved digital investigations (Deshmukh & Ravulakollu, 2024). presented a deep learning-based system that was optimized by utilizing CNN to detect and categorize threats effectively. They achieved an exceptional 95% accuracy rate while drastically reducing training time. These experiments demonstrate how several deep learning techniques can improve IoT intrusion detection systems while striking a balance between security and accuracy.

However, a significant limitation of existing studies is their insufficient evaluation of computational costs and resource constraints typical in IoT environments. Many proposed solutions, while effective in controlled settings, may prove impractical when deployed in real-world IoT systems with limited computational resources, memory, and power. This research addresses these gaps by conducting a comprehensive assessment of computational overhead and resource utilization for IoT intrusion detection implementation, while analyzing the associated security benefits. By explicitly considering the trade-off between security effectiveness and resource consumption, the proposed approach aims to develop solutions that are adaptable across diverse IoT deployments with varying resource constraints. Table 1 presents a comparative analysis between this research and existing studies.

2.3. Security cost–benefit decisions

A cost-benefit analysis supports the evaluation of security implementation options to determine optimal resource investment strategies. Selecting appropriate artificial intelligence solutions for system automation and protection requires complex evaluation criteria (Chaabouni et al., 2019). IoT system threat impacts are typically characterized through probabilistic risk assessments. Research complexity arises from multiple uncertainties including adversarial behavior adaptation, IoT sensor interaction patterns, incomplete feature representation, and evolving attack vectors. IoT networks face both known and unknown vulnerabilities in network traffic. Given the broad scope of IoT network intrusion detection challenges, this research focuses specifically on analyzing the relationship between computational costs and detection accuracy metrics. While other researchers have examined different aspects of IoT security risk quantification (Hemalatha et al., 2021) this work concentrates on measurable parameters of CNN implementations for IoT intrusion detection, including computational overhead and detection performance metrics.

Decision-makers require reliable quantitative data to evaluate implementation and adoption strategies (Street & Olajide, 2021). Security mechanisms impact an organization's financial position through capital expenditure (CAPEX), operational expenses (OPEX), and potential reputational effects. Security features must undergo systematic

Table 1. Comparative analysis of existing works.

Reference	Method	Accuracy	Cost and benefit analysis	Limitations
Hairab et al. (2022)	CNN	91% for mulit-classification	No	Did not assess the computational costs.
Chen et al., (2025)	CNN	88% for multi-classification	No	Less efficient in dynamic intrusion detection scenarios and limited computing resources environment.
Salih and Ibrahim (2023)	LSTM and RNN	84% for multi-class classification	No	Did not consider the limited resources constraints.
Deshmukh and Ravulakollu (2024)	CNN	95% for multi-class classification	No	Did not assess the computational costs.
Altunay and Albayrak (2023)	hybrid CNN + LSTM model	93% for binary classification and 92% for multi-class classification	No	Did not consider the limited resources constraints.
Proposed Method	**CNN**	**93% for binary classification and 96% for multi-class classification**	**Yes**	-

analysis including component breakdown, impact assessment, and evaluation of expected versus actual benefits. This analysis supports evidence-based decision-making for security investments. The conventional approach to security investment evaluation involves calculating the difference between the annualized loss expectancy (ALE) before and after implementing a security control, then subtracting the annual cost of the control implementation and maintenance (Gordon et al., 2020). For IoT network security, decision-makers need performance metrics that capture both the accuracy of traffic classification (benign versus malicious) and the false classification rates. Classification errors occur either through misidentification of traffic types or when the computational overhead of the classification algorithm exceeds practical resource constraints. The research challenge lies in developing algorithms that achieve an optimal balance between classification accuracy, resource efficiency, and system protection while meeting organizational risk tolerance levels.

A cost-benefit analysis requires a structured framework for quantification (Bojanc & Jerman-Blažič, 2008) that defines the scope of financial assessment and establishes measurable categories for costs and benefits. This research focuses specifically on evaluating CNN algorithms for intrusion detection, measuring both computational resource utilization and detection performance metrics. The analysis encompasses implementation overhead, training costs, and runtime resource consumption. The operational costs were calculated per packet processed by the CNN in both binary and multiclass classification modes (Table 3). Additional security mechanism costs and benefits fell outside this research scope.

The CNN's performance can be evaluated through four classification outcomes: true positives (correctly identified malicious packets), true negatives (correctly identified benign packets), false positives (benign packets misclassified as malicious), and false negatives (malicious packets misclassified as benign). These outcomes are quantified using standardized performance metrics detailed in Section 3.2. The metrics include operational resource consumption and computational overhead at both function and code block levels. This comprehensive analysis of computational costs versus detection performance provides decision-makers with quantitative data to evaluate different CNN configurations for IoT intrusion detection implementations. The limitations of this analytical framework are addressed in Section 5.

A systematic review of literature regarding cost evaluation of machine learning applications in IoT intrusion detection systems (IDS) reveals that existing research predominantly focuses on maximizing detection accuracy and optimizing classifier combinations. While these studies have advanced detection capabilities, limited attention has been paid to analyzing the computational costs of different classifiers in resource-constrained IoT environments. Furthermore, there is a notable absence of research that provides comprehensive design metrics and evaluation frameworks for assessing both the efficiency and cost-effectiveness of machine learning-based multiclass intrusion detection systems in IoT contexts. This gap is particularly significant given the resource limitations inherent in IoT deployments. Based on the identified research gap, the following section presents a novel framework that evaluates CNN/LSTM performance through six core architectural components (input processing, convolutional layers, max pooling layers, fully connected layers, softmax activation, and classification output layer) using quantitative metrics including matrix dimensionality, feature vector cardinality, input/output vector sizes, and computational iterations for paired data points

3. Methodology

We focused on developing and evaluating CNN and LSTM architectures to analyze their computational costs and performance trade-offs for IoT intrusion detection using the IoT-23 dataset. The analysis proceeded in two phases: first assessing the CNN's performance for binary classification (benign vs. malicious), then evaluating its multiclass classification capabilities. The Aposemat IoT-23 dataset, collected at the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic, was selected for its comprehensive collection of labeled IoT network traffic containing both benign traffic and real malware infections (Parmisano

et al., 2022). The dataset has been widely adopted in machine learning research for IoT security applications (Ahli et al., 2023) s it satisfies key criteria for IoT malware and intrusion detection research (Strecker et al., 2021). The following subsections detail the neural network architectures, performance metrics, experimental methodology, and data pre-processing techniques.

3.1. CNN architecture

A CNN architecture consists of essential core components and configurable design elements that together determine the network's capabilities (Figure 2). While binary classification represents a simpler architectural design with a single decision boundary, multiclass classification requires more complex architectures that can distinguish between multiple categories. The network's depth and complexity can be adjusted through additional layers to achieve the desired classification performance. The essential components of a minimalistic CNN are:

- Image input array
- Convolution function for feature selection
- ReLU activation function and pooling
- A connected layer
- SoftMax output function
- Determination algorithm

These six components constitute the fundamental building blocks of a CNN architecture designed to classify IoT network traffic as malicious or benign. CNNs optimize computational efficiency through parameter sharing and selective feature extraction, with convolution kernels playing a critical role in both detection accuracy and resource utilization. In multiclass classification implementations, the network's depth is increased through additional hidden layers until sufficient discriminative capacity is achieved. Each additional layer introduces computational overhead, creating a direct relationship between classification complexity and resource costs.

The selection of feature sets representing the target patterns is critical in CNN architecture for computational efficiency. The convolution kernels must effectively process these features at the pixel level, where more precise feature definitions can lead to lower computational overhead. The feature set definition directly impacts both the algorithm's detection accuracy and its operational efficiency. A compact yet discriminative feature set typically provides better cost-effectiveness than larger or less precise alternatives.

In this research, we implemented a minimal CNN architecture utilizing each core component for binary classification of IoT traffic, then extended it with an additional hidden layer for multiclass classification. The algorithm's efficiency is evaluated based on its convergence speed in optimizing feature weights during training, with these learned parameters then applied to subsequent classifications. Detection performance is quantified through the ratio of correct classifications to misclassifications (Equation 1). Using these architectural elements, we developed a CNN capable of both intrusion detection and multiclass categorization of IoT traffic.

Since IoT networks operate under resource constraints with limited computational and energy resources, the CNN architecture was optimized to balance detection performance and resource utilization. The initial CNN design for binary classification (malicious vs. benign traffic) employed a single convolutional layer to minimize

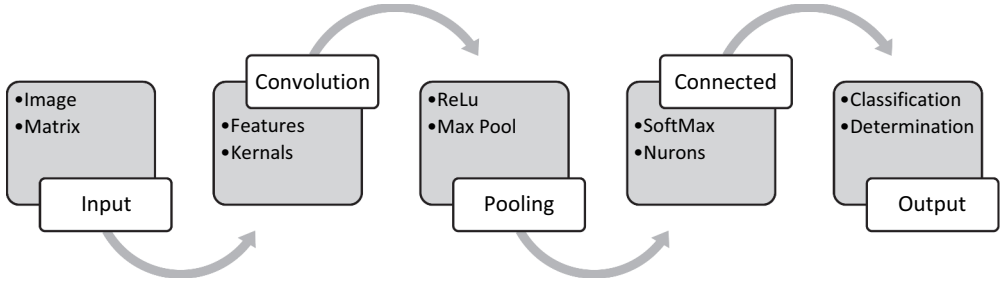

Figure 2. CNN algorithm design.

computational overhead. This layer processes the input features to extract discriminative characteristics such as packet size and connection state patterns. Pooling layers reduce the spatial dimensions of feature maps, decreasing both computational complexity and memory requirements. For binary classification tasks with well-defined feature sets, this shallow architecture proves sufficient for capturing relevant patterns while maintaining efficiency, as the decision boundary between two classes can be learned with fewer parameters compared to multiclass problem (Sengupta et al., 2020). Convolutional kernels extract discriminative features, while ReLU activation functions introduce nonlinearity into the network. Pooling layers reduce spatial dimensions and help prevent overfitting by providing translation invariance. For multiclass classification, the CNN architecture was expanded with an additional hidden layer to capture more complex feature hierarchies. This deeper architecture enables the model to learn sophisticated decision boundaries necessary for distinguishing between multiple traffic categories, as multiclass separation requires more complex feature representations than binary classification (Ruff et al., 2021).

The feature selections occurred as follows: Using the IoT-23 dataset, the following features were chosen to train the CNN model in this study: [“duration,” “orig_bytes,” “resp_bytes,” “missed_bytes,” “orig_pkts,” “orig_ip_bytes,” “resp_pkts,” “resp_ip_bytes,” “proto_icmp,” “proto_tcp,” “proto_udp,” “conn_state_OTH,” “conn_state_REJ,” “conn_state_RSTO,” “conn_state_RSTOS0,” “conn_state_RSTR,” “conn_state_RSTRH,” “conn_state_S0,” “conn_state_S1,” “conn_state_S2,” “conn_state_S3,” “conn_state_SF,” “conn_state_SH,” “conn_state_SHR”]. The selection of these architectural components was optimized for identifying network traffic patterns critical to anomaly detection in IoT environments. These components were specifically chosen based on their effectiveness in processing and classifying IoT network traffic characteristics. The size and frequency of packet transfers are related to key features such as duration, orig_bytes, resp_bytes, orig_pkts, and resp_pkts. These features can reveal anomalous activity, such as data exfiltration, denial-of-service (DoS) attacks, or abnormal usage patterns. Other useful features in differentiating between malicious and benign communications are proto_icmp, proto_tcp, and proto_udp, which correspond to various protocols. Furthermore, the different connection states (i.e., conn_state_* characteristics) offer information about how network sessions end or are terminated, which might help detect activity related to scanning or session-hijacking. After identifying them using domain expertise, we carried out several empirical tests to confirm the relevance of these features. We ensured that every feature was chosen with purpose and made a meaningful contribution to the classification task using correlation analysis (Omuya et al., 2021) and feature importance metrics. Weakly correlated features with traffic type were either eliminated or had their weights changed during the model tuning phase.

3.2. LSTM design

A balance between computational efficiency and model complexity was essential for the LSTM binary classification implementation. The design was simplified for time-series and session-based IoT intrusion detection to ensure consistent performance while maintaining sequential data processing capabilities. To optimize training efficiency and operational costs, the LSTM was configured with minimal hidden units. The LSTM architecture comprises an input layer, multiple hidden LSTM layers, and an output layer. Each LSTM layer contains memory cells with gates (input, forget, and output) that regulate information flow, enabling selective data retention or removal over time. This architecture allows LSTMs to identify long-term dependencies and patterns in sequential data, making them suitable for time-series prediction and sequence analysis tasks. The input data underwent sequential pre-processing, and a minimal hidden layer configuration was implemented for initial model training. The primary computational overhead in this LSTM design stems from the sequential nature of data processing and LSTM unit operations. Like the CNN model, the LSTM demonstrated reliable intrusion detection performance after training on 10,000 .pcap samples. LSTMs typically demand

more computational resources than CNNs due to their recurrent structure. The training process involved 7,500 iterations with a batch size of 256, followed by 2,500 test iterations. The SoftMax function was applied at the output layer for final predictions.

3.3. Cost–benefit analysis metrics

Classification algorithms rarely achieve perfect accuracy due to various factors including incomplete feature representations and inherent uncertainties in IoT network traffic patterns. To evaluate classification performance, we utilized metrics derived from the confusion matrix, which quantifies the relationship between predicted and actual classifications (Equations 1–4) (Hossin & Sulaiman, 2015). When multiclass classification was required, we used Equation 5 to report the statistical accuracy of classifications across all classes (MCC). We expanded Equation 5 by $K = 3$ for the three classes required for our multiclassification. Accuracy, defined as the ratio of correctly classified instances (both true positives and true negatives) to the total number of instances, serves as a primary metric for evaluating the algorithm's classification performance. The *F*1 measure assessed the sample spread, and the MCC measured how closely the multiclass variations related. A score close to 1 for the *F*1 and MCC are ideal results (equation legend: TP = true positive correct malicious classification; TN = true negative correct benign classification; FP = false positive = incorrect malicious classification; FN = incorrect benign classification):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The algorithm's precision was computed as the ratio of correct malicious classifications and the sum of correct and incorrect malicious classifications:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

The recall or true positive rate was computed as the ratio of correct malicious classifications and the sum of correct malicious classifications and incorrect benign classifications:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The *F*1 score is a measure of an ML technique's test accuracy. It calculates accuracy by combining the precision and recall scores computed in Equation 4. The value of the *F*1 score is to acknowledge extreme values in a data set and moderate excessive theoretical normalization (p = precision; R = recall):

$$F1 = 2x\frac{PxR}{P + R} \tag{4}$$

The Mathews Coefficient (Chicco & Jurman, 2020) [44] was used to compare the variables in multiclasses for statistical accuracy, and (5) was computed for each class added:

$$MCC = \frac{c * s - \sum_k^K p_k * t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) * (s^2 - \sum_k^K t_k^2}} \tag{5}$$

k = classes from 1 to K (K = 3in the experiment)
s = number of samples (2,500 in a run)
c = number of samples correctly predicted (2,390 in a run)
t_k = number of times class k truly occurred (328, 492, 1680)
p_k = number of times class k was predicted (297, 466, 1627).

3.4. Experimental design

We run two tests on the computational costs and benefits of the CNN algorithm described in Figure 2. The evaluation proceeded in two phases: initial binary classification of IoT traffic (malicious vs. benign), followed by multiclass classification with three categories using an additional hidden layer. Performance metrics and computational costs were evaluated using the formulas defined in Equations 1-5. The analysis utilized labeled IoT network traffic data from the Aposemat IoT-23 dataset (Garcia et al., 2020). Specifically, the data sets consisted of IoT network traffic captured in. pcap files and labeled malicious, benign, and with seven multiclassification classes for traffic protocols. To use the data sets, we selected two scenarios

for training and testing based on the closest match in packet numbers: benign and malicious scenarios. The former had 21,000 packets, and the latter had 23,000 packets. The literature suggests that 75% of the data should be used for training and 25% for testing (Cui et al., 2018). The experimental design utilized a dataset of 10,000 randomly selected packets, partitioned into 7,500 packets for training and 2,500 for testing. Both CNN and LSTM models were trained on labeled.pcap packets containing benign and malicious traffic samples. For multiclass classification, we selected three protocol categories based on their prevalence in the traffic captures, maintaining the same training-testing split ratio and methodology.

3.5. Data preprocessing and augmentation

The IoT-23 dataset underwent systematic preprocessing to optimize data quality for model training and evaluation. The pre-processing pipeline consisted of several steps: duplicate record removal to prevent bias, filtering of invalid duration values to ensure data integrity, and categorical variable encoding using Label Encoder for machine learning compatibility. The preprocessed dataset was then partitioned using an 80:20 training-testing split ratio, followed by feature normalization using MinMaxScaler to standardize the input distributions (Abd Halim et al., 2020). Preprocessing operations improved the model's overall performance and evaluation accuracy by ensuring that the training data was relevant, clean, and in a format that could be used for model development.

We applied two key data augmentation techniques to the IoT-23 dataset to enhance model performance and reduce overfitting. First, we normalized skewed feature distributions using log scaling transformations, which stabilizes variance and improves learning by creating more uniform feature distributions (Singh et al., 2022). This transformation enhances model accuracy by ensuring comparable feature scales, enabling more effective training. Second, we implemented the MixUp technique, which creates synthetic training examples by linearly interpolating between pairs of samples and their labels (Zou et al., 2023). This approach smooths decision boundaries and increases training data diversity. These augmentation strategies together improved the model's robustness and generalization capabilities while reducing overfitting risks. The MixUp technique specifically helps prevent memorization of training instances and promotes better generalization to novel data patterns.

3.6. Impact of hyperparameter tuning on costs

Model performance and computational costs are significantly influenced by hyperparameter selection, including learning rate, batch size, and number of epochs. We employed the Adam optimizer with a learning rate of 0.001 for model training. Lower learning rates typically provide more stable convergence but require longer training times, while higher rates can accelerate training at the risk of less stable convergence. A batch size of 256 was selected to balance computational efficiency and model performance. Larger batch sizes generally enable faster computation per epoch due to hardware acceleration but require more memory, while smaller batches are more memory-efficient but may need additional epochs for convergence. The selected batch size of 256 optimizes this trade-off between performance and resource utilization. Batch size variations impact both memory usage and training duration, affecting overall computational costs. While increasing the number of epochs can improve model performance through more thorough training, it also extends training time and increases computational overhead. Ten epochs were chosen as an optimal balance between training adequacy and computational efficiency. Our hyperparameter selection reflects a careful consideration of accuracy, training duration, computational cost, and resource utilization requirements.

4. Results

Both binary and multiclass classifiers were designed with minimal complexity to evaluate the fundamental costs and benefits of a baseline CNN architecture. The analysis granularity extended to the code level, with the six core architectural components detailed in Section 3. While additional layers and code complexity could be introduced

to enhance specific capabilities and performance metrics, this minimalistic design approach allows clear visibility of the fundamental components and their contributions. This transparency enables identification of both the baseline benefits for IoT intrusion detection and potential optimization opportunities for future implementations. The following subsections present the experimental results analyzing the computational costs and detection performance of the CNN architecture.

4.1. Binary classification

The CNN binary classification algorithm was designed and trained with emphasis on computational efficiency. We implemented a minimal architecture using optimized computational formulations. This simplified CNN design satisfied performance requirements while maintaining stability, establishing a baseline for evaluating potential architectural enhancements and their associated costs. The input data preprocessing was streamlined by configuring a 3D image format with single-channel depth and grayscale values normalized to the range [0,255]. The array texture obtained was a unique gray scale image representing each.pcap file. Consequently, the minimalistic CNN had a normalized matrix input to the convolution function that identified characteristic features from each matrix. A pooling layer was implemented to reduce computational overhead, while ReLU activation was selected for its efficient computation. The fully connected layer was optimized with minimal neurons, and classification was performed using the computationally efficient SoftMax function, with results collected for performance analysis. The model was trained and evaluated using 10,000 .pcap files as described in Section 3.3, producing stable results that enabled reliable assessment of computational costs and detection performance. Table 2 presents the computational resource utilization metrics of the binary CNN implementation, including per-packet processing costs. The CNN processes data through six sequential architectural components: input layer, convolutional layers, max-pooling layers, fully connected layers, softmax activation, and classification output. Computational costs are determined by multiple factors including training iterations, matrix dimensions, feature map cardinality, and input/output vector sizes. Figures 3 and 4 show the binary CNN design performance benefits during training and testing, respectively, whereas Table 3 reports the costs and benefits metrics.

4.2. Multiclassification

Extending the binary CNN architecture (detailed in Table 2) to support multiclass classification requires additional computational components: an extra convolutional layer, two ReLU activation

Table 2. CNN binary design costs.

	Six architectural components	Cost Determinant	Call Function	Computational Steps (MatLab Code Lines)
CNN Operate	1. Image input	Matrix size	GreyScale()	9
			OpenFile()	14
			ReadLabels()	8
	2. Convolution layer	Feature vector number	Conv(*x*,*y*)	9
			(For Loop)	4 × kernel number
			ReLU()	2
	3. MaxPooling Layer	Input vector size	Pool()	7
			(for loop)	4 × kernel number
			ReLU()	2
	4. Connected layer	Output vector size	Connx()	6
	5. Output SoftMax	Output vector size	SoftMax()	3
	6. Class determination	Two data points	Class()	3
	6.1 Training	Number of runs ($N = 7{,}500$)	Train()	$21 + 12(N) + 6 = 27 + 12(N)$
			Cost per packet	39
	6.2 Test run	Number of runs ($N = 2{,}500$)	Test()	$21 + 12(N) + 6 = 27 + 12(N)$
			Cost per packet	39

Table 3. Comparison of CNN and LSTM in terms of binary metrics.

Method	Accuracy	Precision	Recall	*F*1 Score
CNN	93%	95%	98%	.96
LSTM	91%	92%	95%	.94

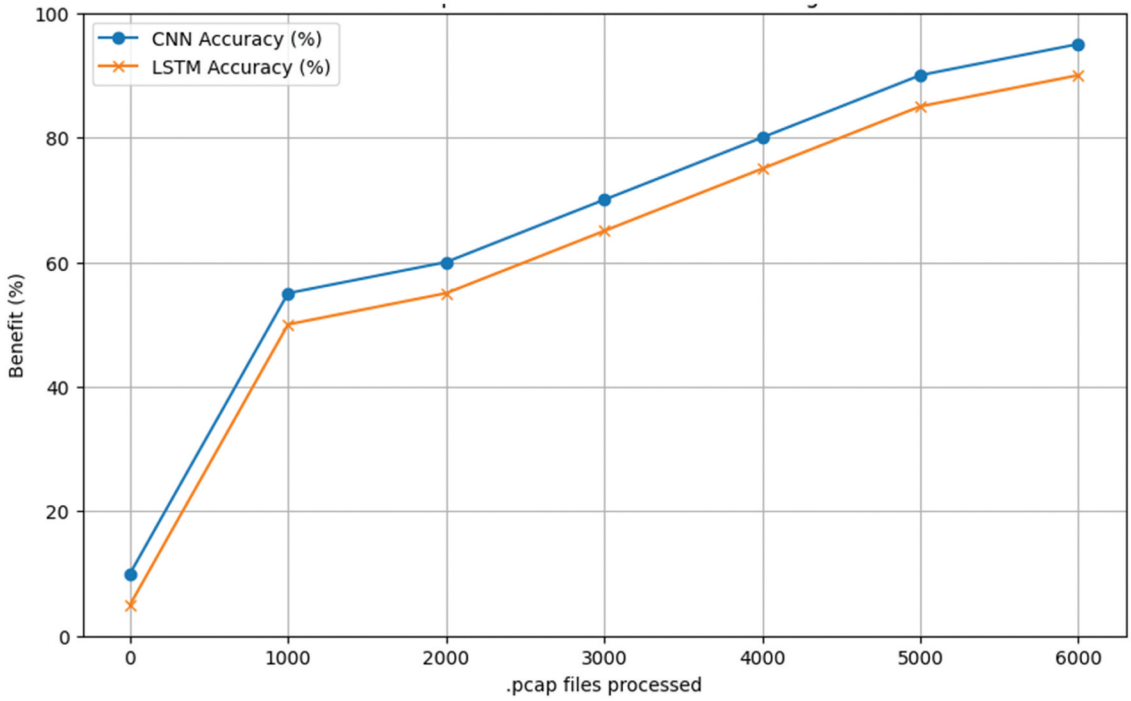

Figure 3. Comparison of binary CNN training benefit rates with LSTM per.Pcap file.

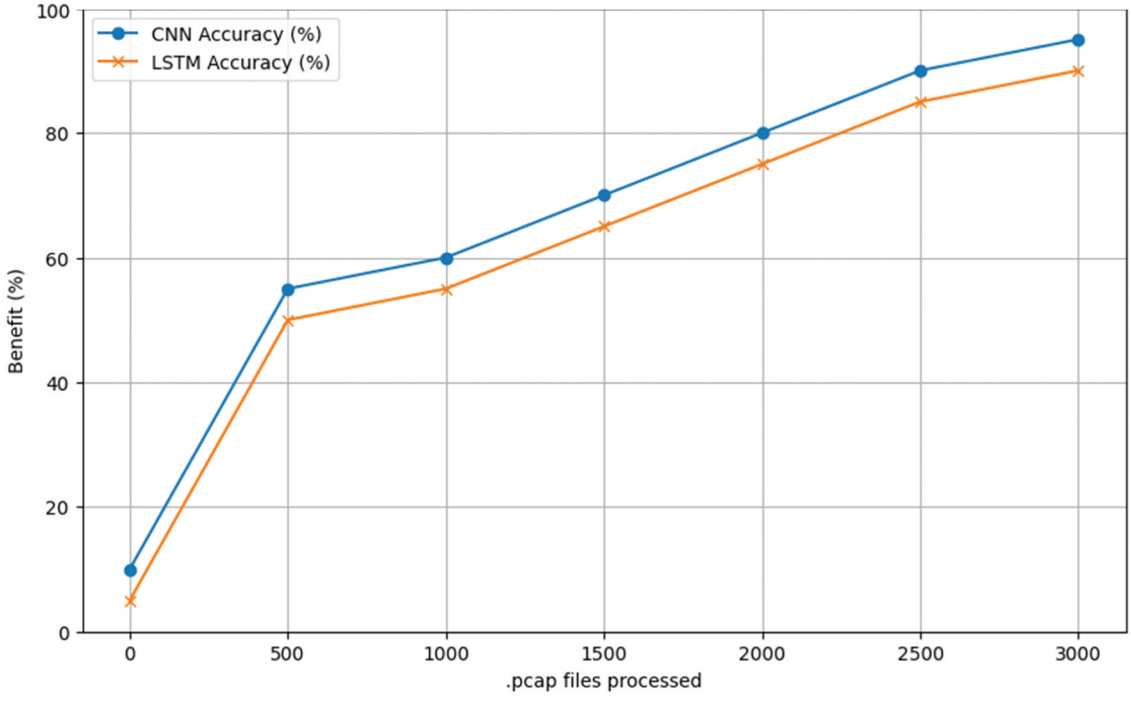

Figure 4. Comparison of binary CNN testing benefit rates with LSTM per.Pcap file.

functions, a pooling layer, 12N packet processing operations, and three metric calculation routines. This architectural expansion increases the per-packet code complexity to 63 lines, representing a 62% increase in computational overhead compared to the binary classification model. Figures 5 and 6 illustrate the multiclass classification performance during training and testing phases

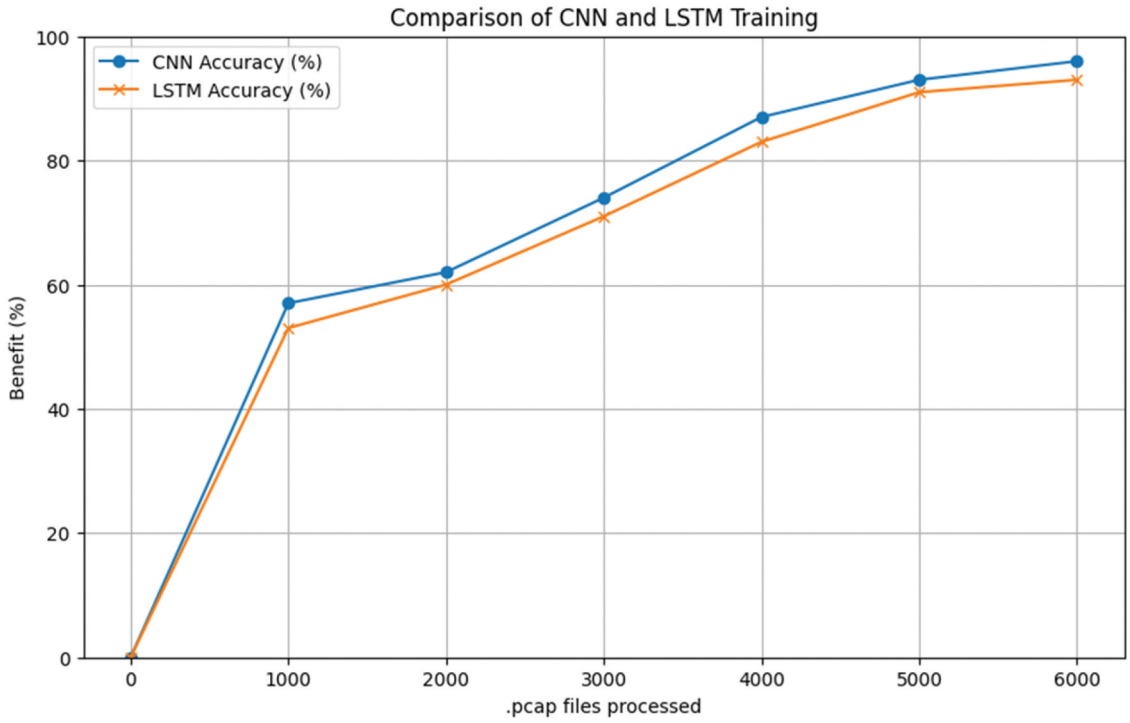

Figure 5. Comparison of multi-class CNN training benefit rates with LSTM per.Pcap file.

respectively, with comprehensive cost-benefit metrics from the evaluation presented in Table 4.

The following data were used from the run data to compute the MCC value:

- k = classes from 1 to $K = 3$ in run
- s = number of samples, 2,500 in run
- c = number of samples correctly predicted is 2,390 in run

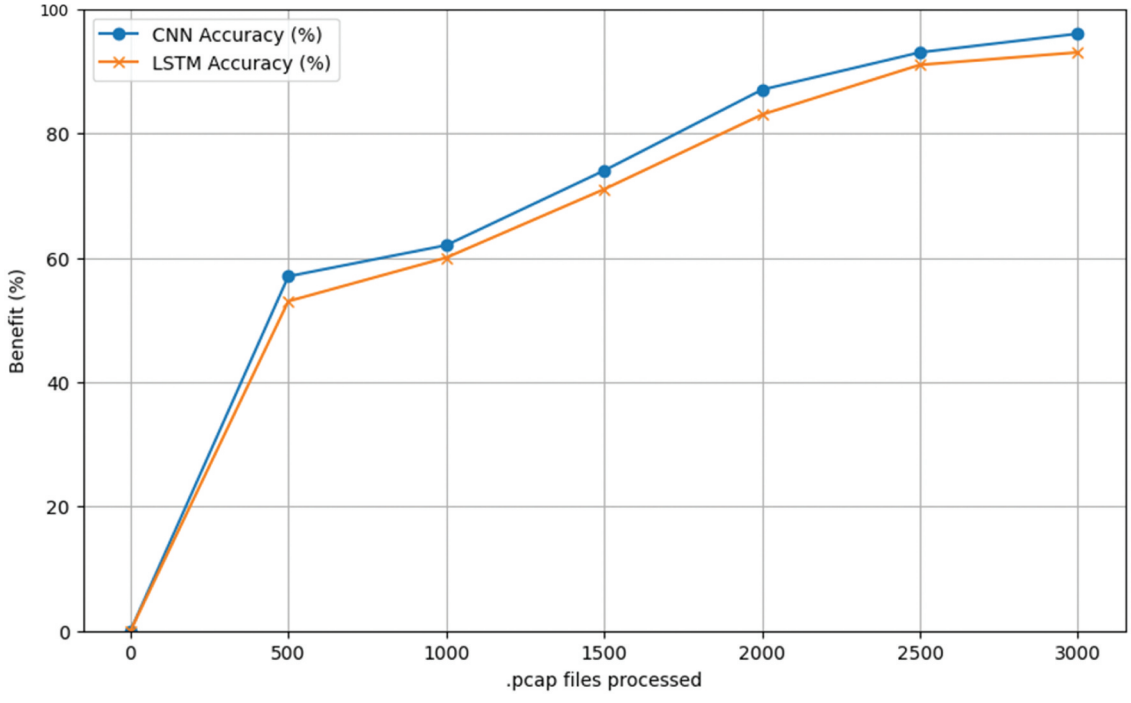

Figure 6. Comparison of multi-class CNN testing benefit rates with LSTM per.Pcap file.

Table 4. Comparison of CNN and LSTM in terms of multiclass metrics.

Method	Accuracy	Precision	Recall	MCC
CNN	96%	99%	99%	.91
LSTM	93%	95%	96%	.87

- t_k = number of times class k truly occurred, by class: 328, 492, 1,680
- p_k = number of times class k was predicted, by class: 297, 466, 1,627

5. Discussion

Table 3 presents the computational costs of the optimized binary classifier, while Figures 3 and 4 demonstrate its performance during training and operational testing, specifically its accuracy in identifying malicious IoT traffic packets. Figure 3 illustrates the training progression, showing initial convergence followed by accelerated optimization of feature weights for binary classification. After processing 7,000 .pcap files, the model achieves autonomous detection with 94% classification accuracy. Figure 4 shows the model achieving 93% classification accuracy after processing 1,500 test packets during operational evaluation. Post-training, the model demonstrates robust adaptation to new patterns while maintaining consistent classification performance. These results validate the technical feasibility of autonomous IoT intrusion detection when misclassification rates remain within acceptable operational risk thresholds. Furthermore, they suggest the CNN's potential as a primary filtering mechanism for high-volume traffic, with flagged packets being redirected to more specialized security mechanisms optimized for lower throughput analysis.

Figure 5 demonstrates that the multiclass CNN architecture with dual convolution layers exhibits slower initial training convergence due to increased classification complexity, but ultimately achieves higher detection accuracy post-training. Figure 6 shows operational performance with slightly delayed convergence compared to the binary classifier (Figure 4), but maintains superior classification capabilities. Table 3 and per-packet cost metrics quantify the computational overhead associated with detection performance. While the binary CNN classifier offers computational efficiency with moderate detection capabilities, the multiclass architecture provides enhanced threat detection at increased computational cost. This presents a key decision point regarding the required security coverage level. Although the binary classifier operates autonomously with minimal overhead, it may not satisfy comprehensive security requirements. The security implementation thus becomes a resource allocation challenge, balancing detection capabilities against computational costs within operational risk tolerances. The 62% computational overhead increase per additional layer for multiclass classification, while providing valuable attack pattern insights and security resource forecasting capabilities, may exceed available computational resources.

The performance metrics (ACC, F1-Score, and MCC) presented in Tables 3 and 4 demonstrate dataset consistency, with CNN performance exhibiting sufficient stability for analytical purposes. Metrics exceeding 0.9 indicate reliable and predictable detection performance suitable for resource planning. Table 3 illustrates that binary classification computational costs, measured in code execution units, scale linearly with packet volume. The CNN's resource utilization demonstrates consistent correlation with architectural complexity and security objectives, following a linear progression. Building upon the binary classification model, the multiclass results in Table 4 suggest that computational costs follow either a step function with each architectural layer addition or an exponential curve as classification complexity increases.

Risk appetite defines an organization's acceptable risk threshold before mitigation measures are required. In IoT security contexts, this threshold is determined by operational requirements and stakeholders' risk tolerance levels, encompassing both operational and financial impact considerations. Risk management frameworks establish specific metrics and tolerance ranges around target risk levels. Table 2 presents 17 IoT sensor types and their vulnerabilities, illustrating how risk context influences tolerance thresholds. For example, smart home lighting sensors have different criticality

levels compared to aircraft pressure sensors or transportation system optical sensors. The CNN implementation achieved accuracy rates of 93% for binary and 96% for multiclass classification, but the significance of these error rates (7% and 4% respectively) must be evaluated within specific operational contexts. Some scenarios may find these error rates unacceptable due to potential attack impacts or recovery costs. For instance, while a DDoS attack on a smoke detector network might be tolerable if the system can alert and recover through software restart, tampering with smart transportation sensors or optical manipulation may pose unacceptable risks. Thus, the suitability of CNN-based IoT intrusion detection systems depends on specific operational contexts and organizational risk management decisions.

The primary advantage of CNN-based IoT intrusion detection, as supported by literature and our findings, is automated processing of large-scale data. Our results demonstrate that once CNN training stabilizes, resource utilization and detection performance become predictable, enabling effective budget modeling. Given the resource constraints at the IoT sensor layer, CNN implementations typically leverage application layer resources. Optimal IoT security implementation therefore requires integration with complementary security mechanisms like lightweight encryption and data masking. Resource assessment is crucial for CNN deployment in IoT environments. This research provides quantitative models and estimates for evaluating individual IoT sensors and systems, as illustrated in Figure 1, supporting informed decision-making regarding CNN implementation. While optimal security configurations may exceed available resource constraints, rational trade-offs can lead to feasible solutions. Realizing CNN benefits in IoT security requires alignment between performance metrics, cost models, potential loss scenarios, and system-specific resource and operational constraints.

Scalability is critical for maintaining computational efficiency as CNN models process increasing data volumes. While larger datasets typically demand greater computational resources and training time, potentially straining hardware and operational costs, various optimization techniques can address these challenges. Model pruning reduces parameter count, while quantization decreases computational precision, both enhancing scalability without significantly impacting detection accuracy. Hardware acceleration through Graphics Processing Units (GPUs) can substantially improve training efficiency and throughput. Implementing these optimization strategies in combination enables the model to scale effectively with growing data volumes while maintaining resource efficiency.

Implementation of CNN models in IoT environments requires consideration of several critical factors. The model must be optimized for real-time processing through specialized hardware acceleration or model compression techniques to minimize inference latency. Resource constraints of IoT edge devices necessitate model optimization through pruning, quantization, or lightweight architectures. Additionally, the model must adapt to evolving attack patterns, requiring regular updates and retraining mechanisms to maintain detection effectiveness.

5.1. Comparison with the LSTM model

Comparative analysis revealed that the CNN architecture demonstrated superior performance and computational efficiency compared to the LSTM model. The CNN achieved higher accuracy, recall, and precision metrics in both binary and multiclass classification tasks. Its ability to process spatial hierarchies in parallel contributed to improved performance metrics while reducing computational overhead. Although the LSTM model is designed for sequential pattern recognition, its recurrent architecture requires greater computational resources, resulting in extended training and inference times. Despite optimizing the LSTM through minimal hidden units and efficient input representation, the CNN maintained better resource efficiency. The LSTM's sequential processing nature introduced additional computational overhead, reducing its effectiveness compared to the CNN architecture. For this specific intrusion detection application, the CNN proved more suitable due to its superior detection performance and lower computational requirements.

6. Conclusion

The implementation of artificial intelligence for IoT network security requires careful evaluation of resource costs against security benefits. The rapid proliferation of IoT devices necessitates cost-effective security solutions that balance protection against resource constraints, minimizing potential vulnerabilities and associated losses. Addressing the research gap in IoT security resource evaluation, we developed and analyzed simplified CNN architectures for binary and multiclass intrusion detection. The research introduces a novel framework evaluating six core architectural components (input processing, convolutional layers, max pooling layers, fully connected layers, softmax activation, and classification output) using quantitative metrics (matrix dimensionality, feature vector cardinality, input/output vector sizes, computational iterations, and per-packet processing costs). Results demonstrate that trained CNNs can efficiently process high-volume IoT traffic within predictable resource constraints, though detection accuracy requirements must align with application-specific risk tolerances.

While this research advances understanding of cost-benefit trade-offs in AI-based intrusion detection, several limitations warrant consideration. The evaluation's scope, limited to a single dataset and two classifier architectures, may not fully represent the heterogeneous IoT ecosystem. Future research should expand along three critical dimensions: diverse ML classifiers beyond CNN and LSTM, heterogeneous IoT environments with varying resource constraints, and comprehensive attack vectors including emerging threats. This expanded scope would enable more robust validation of the resource evaluation framework, particularly in large-scale industrial deployments. Additionally, the proposed metrics and evaluation methodology can be extended to assess classifier performance and resource utilization across different IoT contexts, enabling systematic comparison of security solutions while considering computational overhead and detection efficacy.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

We thank Rabdan Academy for funding the research presented in this paper.

Data availability statement

Dataset available on request.

References

Abd Halim, K. N., Jaya, A. M., & Fadzil, A. F. A. (2020). Data pre-processing algorithm for neural network binary classification model in bank tele-marketing. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *9*(3), 272–277. https://doi.org/10.35940/ijitee.C8472.019320

Ahli, A., Raza, A., Akpinar, K. O., & Akpinar, M. (2023). *Binary and multi-class classification on the IoT-23 dataset* [Paper presented]. 2023 Advances in Science and Engineering Technology International Conferences (ASET), UAE.

Alam, M., & Khan, I. R. (2022). *Cyber-physical attacks and IoT intelligent cyber-physical systems security for industry 4.0*. Chapman and Hall/CRC.

Al-Awami, S. H., Al-Aty, M. M., & Al-Najar, M. F. (2023). *Comparison of IoT architectures based on the seven essential characteristics* [Paper presented]. 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Benghazi, Libya.

Alhowaide, A., Alsmadi, I., & Tang, J. (2021). Towards the design of real-time autonomous IoT NIDS. *Cluster Computing*, *26*(5), 1–14. https://doi.org/10.1007/s10586-021-03231-5

Alladi, T., Chamola, V., Sikdar, B., & Choo, K.-K. R. (2020). Consumer IoT: Security vulnerability case studies and solutions. *IEEE Consumer Electronics Magazine*, *9*(2), 17–25. https://doi.org/10.1109/MCE.2019.2953740

Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN+ LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, *38*, 101322. https://doi.org/10.1016/j.jestch.2022.101322

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, *9*(7), 1177. https://doi.org/10.3390/electronics9071177

Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. *Procedia Computer Science*, *185*, 239–247. https://doi.org/10.1016/j.procs.2021.05.025

Baig, D., Khan, M. U., Dancey, D., Abbas, A., Ali, M., & Nawaz, R. (2021). *Malware detection and classification along with trade-off analysis for number of features, feature types, and speed* [Paper presented]. 2021 International

Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan.

Baraneetharan, D. E. (2020). Role of machine learning algorithms intrusion detection in WSNs: A survey. *Journal of Information Technology and Digital World*, *2*(3), 161–173. https://doi.org/10.36548/jitdw.2020.3.004

Baziyad, H., Kayvanfar, V., & Kinra, A. (2022). The internet of things-an emerging paradigm to support the digitalization of future supply chains. In *The digital supply chain* (pp. 61–76). Elsevier.

Birman, Y., Hindi, S., Katz, G., & Shabtai, A. (2022). Cost-effective ensemble models selection using deep reinforcement learning. *Information Fusion*, *77*, 133–148. https://doi.org/10.1016/j.inffus.2021.07.011

Bojanc, R., & Jerman-Blažič, B. (2008). An economic modelling approach to information security risk management. *International Journal of Information Management*, *28*(5), 413–422. https://doi.org/10.1016/j.ijinfomgt.2008.02.002

Cavusoglu, H., Cavusoglu, H., & Raghunathan, S. (2004). Economics of ITSecurity management: Four improvements to current security practices. *Communications of the Association for Information Systems*, *14*(1), 3. https://doi.org/10.17705/1CAIS.01403

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials*, *21*(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Chen, H., Wang, Z., Yang, S., Luo, X., He, D., & Chan, S. (2025). Intrusion detection using synaptic intelligent convolutional neural networks for dynamic internet of things environments. *Alexandria Engineering Journal*, *111*, 78–91. https://doi.org/10.1016/j.aej.2024.10.014

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 1–13. https://doi.org/10.1186/s12864-019-6413-7

Čolaković, A., & Hadžialić, M. (2018). Internet of things (IoT): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, *144*, 17–39. https://doi.org/10.1016/j.comnet.2018.07.017

Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G.-G., & Chen, J. (2018). Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics*, *14*(7), 3187–3196. https://doi.org/10.1109/TII.2018.2822680

Deshmukh, A., & Ravulakollu, K. (2024). An efficient CNN-based intrusion detection system for IoT: Use case towards cybersecurity. *Technologies*, *12*(10), 203. https://doi.org/10.3390/technologies12100203

Dey, P., Chaulya, S. K., & Kumar, S. (2021). Hybrid CNN-LSTM and IoT-based coal mine hazards monitoring and prediction system. *Process Safety and Environmental Protection*, *152*, 249–263.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, 100379. https://doi.org/10.1016/j.cosrev.2021.100379

Eke, H. N., Petrovski, A., & Ahriz, H. (2019, September). The use of machine learning algorithms for detecting advanced persistent threats. *Proceedings of the 12th international conference on security of information and networks* (pp. 1–8).

Elrawy, M. F., Awad, A. I., & Hamed, H. F. (2018). Intrusion detection systems for IoT-based smart environments: A survey. *Journal of Cloud Computing*, *7*(1), 1–20. https://doi.org/10.1186/s13677-018-0123-6

Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020). *IoT-23: A labeled dataset with malicious and benign IoT network traffic (version 1.0.0)*. Zenodo.

Ghirardello, K., Maple, C., Ng, D., & Kearney, P. (2018). *Cyber security of smart homes: Development of a reference architecture for attack surface analysis* [Paper presented]. Living in the Internet of Things: Cybersecurity of the IoT-2018.

Gibert, D., Mateu, C., & Planes, J. (2020). HYDRA: A multimodal deep learning framework for malware classification. *Computers & Security*, *95*, 101873. https://doi.org/10.1016/j.cose.2020.101873

Gibert, D., Mateu, C., Planes, J., & Vicens, R. (2019). Using convolutional neural networks for classification of malware represented as images. *Journal of Computer Virology and Hacking Techniques*, *15*(1), 15–28. https://doi.org/10.1007/s11416-018-0323-0

Gordon, L. A., & Loeb, M. P. (2006). Budgeting process for information security expenditures. *Communications of the ACM*, *49*(1), 121–125. https://doi.org/10.1145/1107458.1107465

Gordon, L. A., Loeb, M. P., & Zhou, L. (2020). Integrating cost–benefit analysis into the NIST cybersecurity framework via the Gordon–Loeb Model. *Journal of Cybersecurity*, *6*(1), tyaa005. https://doi.org/10.1093/cybsec/tyaa005

Hairab, B. I., Elsayed, M. S., Jurcut, A. D., & Azer, M. A. (2022). Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks. *Institute of Electrical and Electronics Engineers Access*, *10*, 98427–98440. https://doi.org/10.1109/ACCESS.2022.3206367

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on IoT security: Application areas, security threats, and solution architectures. *Institute of Electrical and Electronics Engineers Access*, *7*, 82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045

Hemalatha, J., Roseline, S. A., Geetha, S., Kadry, S., & Damaševičius, R. (2021). An efficient densenet-based deep learning model for malware detection. *Entropy*, *23*(3), 344. https://doi.org/10.3390/e23030344

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and

future challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686–1721. https://doi.org/10.1109/COMST.2020.2986444

Kim, K., & Aminanto, M. E. (2017). *Deep learning in intrusion detection perspective: Overview and further challenges* [Paper presented]. 2017 International Workshop on Big Data and Information Security (IWBIS), Jakarta.

Kuzlu, M., Fair, C., & Guler, O. (2021). Role of artificial intelligence in the internet of things (IoT) cybersecurity. *Discover Internet of Things*, *1*(1), 1–14. https://doi.org/10.1007/s43926-020-00001-4

Lee, I., & Lee, K. (2015). The internet of things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), 431–440. https://doi.org/10.1016/j.bushor.2015.03.008

Liu, L., Wang, B.-S., Yu, B., & Zhong, Q.-X. (2017). Automatic malware classification and new malware detection using machine learning. *Frontiers of Information Technology & Electronic Engineering*, *18*(9), 1336–1347. https://doi.org/10.1631/FITEE.1601325

Liu, P., Wang, J., Sangaiah, A. K., Xie, Y., & Yin, X. (2019). Analysis and prediction of water quality using LSTM deep neural networks in IoT environment. *Sustainability*, *11*(7), 2058.

Mendonça, R. V., Teodoro, A. A., Rosa, R. L., Saadi, M., Melgarejo, D. C., Nardelli, P. H., & Rodríguez, D. Z. (2021). Intrusion detection system based on fast hierarchical deep convolutional neural network. *Institute of Electrical and Electronics Engineers Access*, *9*, 61024–61034. https://doi.org/10.1109/ACCESS.2021.3074664

Mijalkovic, J., & Spognardi, A. (2022). Reducing the false negative rate in deep learning based network intrusion detection systems. *Algorithms*, *15*(8), 258. https://doi.org/10.3390/a15080258

Mohanta, B. K., Jena, D., Satapathy, U., & Patnaik, S. (2020). Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of Things*, *11*, 100227. https://doi.org/10.1016/j.iot.2020.100227

Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion detection system using feature extraction with machine learning algorithms in IoT. *Journal of Sensor and Actuator Networks*, *12*(2), 29. https://doi.org/10.3390/jsan12020029

Najar-Pacheco, J. C., Bohada-Jaime, J. A., & Rojas-Moreno, W. Y. (2019). Vulnerabilities in the internet of things. *Visión Electrónica*, *13*(2), 312–321. https://doi.org/10.14483/22484728.15163

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. *IEEE Communications Surveys & Tutorials*, *21*(3), 2702–2733. https://doi.org/10.1109/COMST.2019.2910750

Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. *Computers & Security*, *77*, 871–885. https://doi.org/10.1016/j.cose.2018.04.005

Nicho, M., & Girija, S. (2021, December). IoTVT model: A model mapping IoT sensors to IoT vulnerabilities and threats. *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)* (pp. 123–129). IEEE.

Omuya, E. O., Okeyo, G. O., & Kimwele, M. W. (2021). Feature selection for classification using principal component analysis and information gain. *Expert Systems with Applications*, *174*, 114765. https://doi.org/10.1016/j.eswa.2021.114765

Parmisano, A., Garcia, S., & Erquiaga, M. J. (2022). *Aposemat IoT-23: A labeled dataset with malicious and benign IoT network traffic. From Stratosphere Laboratory, AIC group, FEL, CTU university*. https://mcfp.felk.cvut.cz/publicDatasets/IoT-23-Dataset/

Rodrigues, B., Franco, M., Parangi, G., & Stiller, B. (2019). *SEConomy: A framework for the economic assessment of cybersecurity* [Paper presented]. Economics of Grids, Clouds, Systems, and Services: 16th International Conference, GECON 2019, Leeds, UK. September 17–19, 2019

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Müller, K. R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, *109*(5), 756–795.

Salih, K. M. M., & Ibrahim, N. B. (2023). *Enhancing IoT forensics through deep learning: Investigating cyber-attacks and analyzing big data for improved security measures* [Paper presented]. 2023 4th International Conference on Big Data Analytics and Practices (IBDAP), Bangkok, Thailand.

Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., & Stiller, B. (2022). Landscape of IoT security. *Computer Science Review*, *44*, 100467. https://doi.org/10.1016/j.cosrev.2022.100467

Sengupta, J., Ruj, S., & Bit, S. D. (2020). A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. *Journal of Network and Computer Applications*, *149*, 102481. https://doi.org/10.1016/j.jnca.2019.102481

Serror, M., Hack, S., Henze, M., Schuba, M., & Wehrle, K. (2020). Challenges and opportunities in securing the industrial internet of things. *IEEE Transactions on Industrial Informatics*, *17*(5), 2985–2996. https://doi.org/10.1109/TII.2020.3023507

Sethi, P., & Sarangi, S. R. (2017a). Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, *2017*, 1–25. https://doi.org/10.1155/2017/9324035

Sethi, P., & Sarangi, S. R. (2017b). Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, *2017*, 1–25. https://doi.org/10.1155/2017/9324035

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM)

network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Shetty, N., Schwartz, G., Felegyhazi, M., & Walrand, J. (2010). *Competitive cyber-insurance and internet security* [Paper presented]. Economics of information security and privacy.

Singh, A., Amutha, J., Nagar, J., Sharma, S., & Lee, C.-C. (2022). Lt-fs-id: Log-transformed feature learning and feature-scaling-based machine learning algorithms to predict the k-barriers for intrusion detection using wireless sensor network. *Sensors (Switzerland)*, *22*(3), 1070. https://doi.org/10.3390/s22031070

Strecker, S., Dave, R., Siddiqui, N., & Seliya, N. (2021). A modern analysis of aging machine learning based IOT cybersecurity methods. *Journal of Computer Sciences and Applications*, *9*(1), 16–22. *arXiv preprint arXiv: 2110.07832*. https://doi.org/10.12691/jcsa-9-1-2

Street, J., & Olajide, F. (2021). A cost-benefit analysis of information security mitigation methods for ORVIs. *Journal of Internet Technology and Secured Transactions*, *9* (1), 747–755. https://doi.org/10.20533/jitst.2046.3723.2021.0092

Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of internet of things (IoT): A survey. *Journal of Network and Computer Applications*, *161*, 102630. https://doi.org/10.1016/j.jnca.2020.102630

Thamilarasu, G., & Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. *Sensors (Switzerland)*, *19*(9), 1977. https://doi.org/10.3390/s19091977

Wang, Z., Ghaleb, F. A., Zainal, A., Siraj, M. M., & Lu, X. (2024). An efficient intrusion detection model based on convolutional spiking neural network. *Scientific Reports*, *14*(1), 7054. https://doi.org/10.1038/s41598-024-57691-x

Ye, Y., Chen, L., Hou, S., Hardy, W., & Li, X. (2018). DeepAM: A heterogeneous deep learning framework for intelligent malware detection. *Knowledge and Information Systems*, *54* (2), 265–285. https://doi.org/10.1007/s10115-017-1058-9

Yu, M., Xu, F., Hu, W., Sun, J., & Cervone, G. (2021). Using long short-term memory (LSTM) and Internet of Things (IoT) for localized surface temperature forecasting in an urban environment. *Institute of Electrical and Electronics Engineers Access*, *9*, 137406–137418.

Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. (2022). Comparative research on network intrusion detection methods based on machine learning. *Computers & Security*, *121*, 102861. https://doi.org/10.1016/j.cose.2022.102861

Zhang, H., Wu, C. Q., Gao, S., Wang, Z., Xu, Y., & Liu, Y. (2018). *An effective deep learning based scheme for network intrusion detection* [Paper presented]. 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China.

Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., & Yu, B. (2019). Recent advances in convolutional neural network acceleration. *Neurocomputing*, *323*, 37–51. https://doi.org/10.1016/j.neucom.2018.09.038

Zoppi, T., Gharib, M., Atif, M., & Bondavalli, A. (2021). Meta-learning to improve unsupervised intrusion detection in cyber-physical systems. *ACM Transactions on Cyber-Physical Systems (TCPS)*, *5*(4), 1–27. https://doi.org/10.1145/3467470

Zou, D., Cao, Y., Li, Y., & Gu, Q. (2023). *The benefits of mixup for feature learning* [Paper presented]. International Conference on Machine Learning, Hawaii, USA.