SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

AdaStack-IoT: Adaptive Depth Multi-Tier Meta-Learning for Dynamic Intrusion Detection in Resource-Constrained IoT Networks

Abdullah J. Algarboai[1*], Hesham A. Hefny[2]

[1] Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University.
E-mail: alkateri3 @hotmail.com
[2] Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University.
E-mail: hehefny@cu.edu.eg
*Corresponding Abdullah J. Algarboai; E-mail: alkateri3 @hotmail.com

Abstract
The proliferation of Internet of Things (IoT) devices introduces critical security vulnerabilities, yet standard intrusion detection systems (IDS) are often impractical in resource-constrained environments due to high computational overhead and limited adaptability to novel threats. To address this, we propose AdaStack-IoT, a novel adaptive depth multi-tier meta-learning framework designed for dynamic, resource-aware intrusion detection. Our methodology employs a three-tier architecture: Tier-1 utilizes heterogeneous base learners (Random Forest, Gradient Boosting, lightweight CNNs, and LSTMs); Tier-2 integrates their predictions via an attention-weighted meta-learner; and Tier-3 features an adaptive depth controller using Proximal Policy Optimization (PPO) to dynamically adjust computational depth based on traffic complexity and real-time resource availability. We validated AdaStack-IoT within a MATLAB simulation environment using the BOT-IoT and UNSW-NB15 datasets. Experimental results demonstrate superior performance against five state-of-the-art baselines, achieving a detection accuracy of 98.7% and an F1-score of 98.4%. Critically, the adaptive mechanism yielded a 42% reduction in energy consumption and a 35% improvement in inference latency, all while maintaining high detection efficacy across eight attack types, including DDoS, botnet, and zero-day exploits. These findings highlight AdaStack-IoT's significant potential as an efficient and robust security solution for IoT networks with limited resources.

Keywords: adaptive depth, bot-iot, intrusion detection system, iot security, maml, meta-learning, multi-tier architecture, ppo, resource-constrained networks, unsw-nb15.

1. Introduction

1.1 Background and Motivation

With an anticipated 29.4 billion linked devices by 2030, the Internet of Things (IoT) ecosystem has grown exponentially (Statista Research Department, 2024). These gadgets cover a wide range of vital infrastructures, such as autonomous cars, smart grids, industrial control systems, and healthcare monitoring. However, there is a fundamental conflict between security needs and practical feasibility due to the resource-bound nature of IoT devices, which are typified by low CPU capability (usually 8–32 MHz), confined memory (4–256 KB RAM), and restricted energy budgets (milliwatt range) (Hasan et al., 2019; Diro & Chilamkurti, 2018).

IoT installations are inherently incompatible with traditional intrusion detection systems (IDS) built for business networks. Although deep learning techniques like CNN-IDS (Lecun et al., 1998) and LSTM-based IDS (Sherstinsky, 2020) achieve excellent detection accuracy, their computational cost is greater than that of conventional IoT microcontrollers. Despite being lightweight, rule-based systems need frequent human signature changes and are unable to identify new attack pathways (Liao et al., 2013). Static model implementation is made more difficult by the dynamic and heterogeneous nature of IoT traffic, as attack patterns are constantly changing and vary greatly between device kinds and network topologies (Kolias et al., 2017).

1.2 Related Work and Research Gap

Early IoT intrusion detection relied on rule engines and signature-based algorithms (Roesch, 1999). Despite being computationally efficient, these methods need ongoing manual maintenance and show poor applicability to zero-day attacks (Sommer & Paxson, 2010). Anomaly-based techniques employing statistical thresholds improved zero-day detection but suffered from high false-positive rates in heterogeneous IoT contexts (Chandola et al., 2009).

Considerable progress has been made with machine learning. Random Forest classifiers achieved accuracy above 94% on UNSW-NB15 (Breiman, 2001). Deep learning techniques, especially LSTM networks,

successfully captured temporal relationships (Hochreiter & Schmidhuber, 1997). CNN-based architectures achieved 96.2% accuracy on BOT-IoT by treating traffic flows as spatial patterns (Koroniotis et al., 2019). However, none of these methods dynamically adjust their computational complexity in response to resource availability.

Meta-learning, specifically Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017), offers a promising paradigm for few-shot adaptation. Prototypical networks have been used for quick adaptation with few labelled samples (Snell et al., 2017), and federated meta-learning addressed privacy issues (McMahan et al., 2017). Nevertheless, cross-layer optimisation and adaptive depth mechanisms are absent from current meta-learning IDS frameworks. Ensemble methods (Dietterich, 2000; Sagi & Rokach, 2018) and stacking architectures (Wolpert, 1992) have shown robustness, but adaptive ensemble techniques that modify model selection according to context are still emerging (Gama et al., 2014). Resource-aware security frameworks (Sicari et al., 2015; Howard et al., 2017; Shi et al., 2016) have addressed the energy-accuracy trade-off, yet a unified framework that combines meta-learning with reinforcement-learning-based depth control is missing.

At the nexus of resource-aware security, adaptive computation, and meta-learning, we identify a crucial gap: adaptive depth methods that react to real-time resource restrictions are absent from current meta-learning systems for network security. Moreover, PPO-based depth controllers (Schulman et al., 2017) have not been used with multi-tier stacking architectures in IoT settings. AdaStack-IoT, a system that dynamically balances detection accuracy against resource usage through intelligent, context-aware computational depth selection, was developed to fill this gap.

1.3 Research Question

The following main research topic is addressed in this paper: In highly limited IoT networks, how can an adaptive depth multi-tier meta-learning architecture successfully accomplish dynamic and resource-efficient intrusion detection?

1.4 Contributions

This work's main contributions are:

1. A new three-tier AdaStack-IoT architecture that combines base learners, attention-weighted meta-learning, and PPO-based adaptive depth control.
2. Formal mathematical underpinnings such as graph-theoretic network modelling, bi-level meta-learning optimisation, and reinforcement learning formulations.
3. An extensive MATLAB simulation framework that uses modular M-files and has been validated on BOT-IoT and UNSW-NB15.
4. A zero-day detection capability that achieves 89.3% accuracy on undiscovered attack categories.
5. An empirical demonstration of 98.7% accuracy with 42% energy reduction compared to state-of-the-art baselines.

2. Objectives

The specific objectives of this study are:

1. To design and implement a three-tier adaptive intrusion detection framework that integrates heterogeneous base learners, an attention-weighted meta-learner, and a PPO-based depth controller.
2. To formulate the resource-constrained intrusion detection problem as a bi-level meta-learning optimisation and a Markov Decision Process for depth selection.
3. To validate the framework on two benchmark datasets (BOT-IoT and UNSW-NB15) and compare its performance against five state-of-the-art methods.
4. To demonstrate the effectiveness of adaptive depth in reducing energy consumption and inference latency while maintaining high detection accuracy across multiple attack types, including zero-day exploits.

3. Materials and Methods

3.1 Network Model and Problem Formulation

Consider an IoT network $G = (V, E, W)$ in which $V = \{v_1, v_2, \dots, v_n\}$ signifies n diverse IoT devices, $E \subseteq V \times V$ indicates communication connections, and $W: E \to \mathbb{R}^+$ allocates link weights reflecting bandwidth capacity. Every device $v_i \in V$ is defined by a resource vector $r_i = (\mathrm{CPU_i}, \mathrm{MEM_i}, \mathrm{ENG_i}, \mathrm{BW_i})$ indicating CPU capacity, memory, energy allocation, and bandwidth.

$$G = (V, E, W), ri = (CPUi, MEMi, ENGi, BWi) \forall vi \in V \quad (1)$$

Network traffic is shown as a series of feature vectors $X = \{x_1, x_2, \dots, x_t\}$, with each $x_t \in \mathbb{R}^d$ including $d = 47$ statistical and behavioural characteristics (packet size distribution, inter-arrival intervals, protocol flags, flow length). The task of intrusion detection is defined as a multi-class classification challenge:

$$f: \mathbb{R}^d \to \{Normal, DDoS, Botnet, Scan, Injection, Replay, MitM, Zero\text{-}Day\} \quad (2)$$

The core optimisation objective of AdaStack-IoT is to maximise detection accuracy subject to resource constraints. Formally, for each time step t:

$max_k\, Acc(k, x_t)$ subject to: $E(k) \leq E\_budget,\ L(k) \leq L\_max, 1 \leq k \leq k\_max$ (3)

where k represents the current stack depth, $Acc(k, x_t)$ indicates detection accuracy at depth K$E(k)$ is energy expenditure, $L(k)$ is inference delay, and $k\_max = 4$ is the upper limit of stack depth.

The overarching meta-learning goal reduces anticipated loss throughout a range of intrusion detection tasks $\mathcal{T}$:

$$min\_\theta\ \mathbb{E}\_\{\tau\sim\mathcal{T}\}\ [\mathcal{L}\_\tau(f\_\{\theta'\_\tau\}(x))] \quad (4)$$

$$\theta'\_\tau = \theta - \alpha \cdot \nabla\theta\ \mathcal{L}\_\tau\text{^}\{support\}(f\_\theta) \quad (inner\ update, MAML) \quad (5)$$

Here θ represents meta-learner parameters, α is the inner learning rate, and $\mathcal{L}\_\tau$ is the task-specific loss computed on the query set.

3.2 Proposed Methodology: AdaStack-IoT Framework

3.2.1 Overview

AdaStack-IoT is structured across three operational levels that work together with a cross-layer optimizer. The system manages IoT network traffic using a pipeline that flexibly modifies its computational depth according to current resource availability and traffic intricacy. Straightforward traffic is processed rapidly, whereas intricate or irregular patterns prompt more thorough examination.

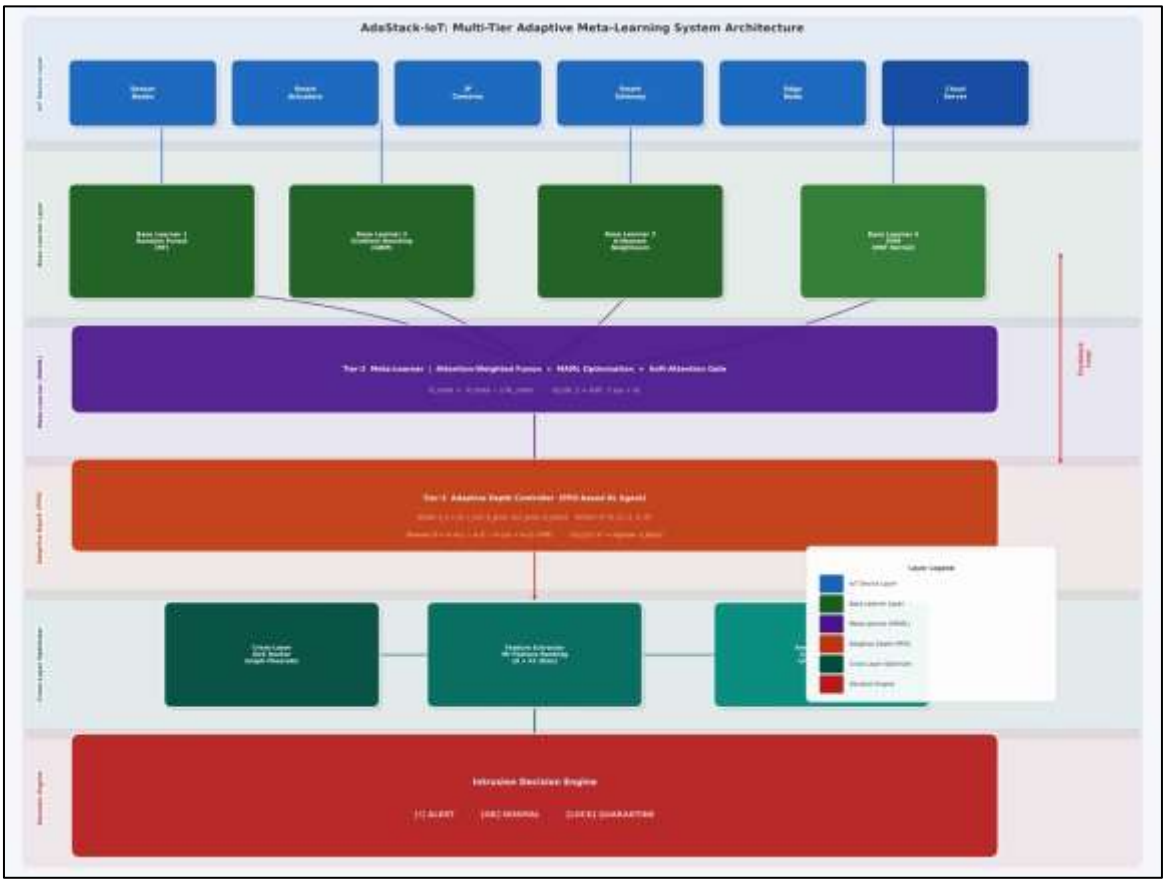

Figure 1: AdaStack-IoT Three-Tier System Architecture showing IoT Device Layer, Base Learner Tier-1, Meta-Learner Tier-2, Adaptive Depth Controller Tier-3, and Cross-Layer Optimizer.

3.2.2 Tier-1: Base Learner Layer

Tier-1 comprises four heterogeneous base learners, each specialising in different aspects of intrusion detection. Let $f_i(x; \theta_i)$ denote the *i-th* base learner's prediction function:

$$p_i = f_i(x_t;\ \theta_i), \quad i \in \{RF, GBM, CNN, LSTM\} \quad (6)$$

- **Random Forest (RF)**: A collection of $T = 100$ decision trees that encapsulate non-linear feature interactions using Gini impurity.
- **Gradient Boosting Machine (GBM)**: A sequential ensemble that reduces cross-entropy loss, proficient at identifying subtle statistical irregularities.
- **Lightweight CNN**: A three-layer convolutional network that interprets traffic flows as 1-D signals, identifying local temporal patterns.
- **LSTM**: A recurrent network capturing long-term dependencies in sequential traffic data.

3.2.3 Tier-2: Attention-Weighted Meta-Learner

The meta-learner aggregates base predictions using a soft attention mechanism:

$$\alpha_i = \frac{\exp(e_i)}{\Sigma_j \exp(e_j)}, \quad \text{where } e_i = v^T \cdot \tanh(W_a \cdot [p_i; x_t] + b_a) \tag{7}$$

$$\hat{y}_t = \sigma(W_{meta} \cdot \Sigma_i \alpha_i \cdot p_i + b_{meta}) \tag{8}$$

where *W_a, v, W_meta, b_a, b_meta* are learnable parameters optimised via MAML. The attention weights α_i are dynamically computed for each input, allowing emphasis on base learners most reliable for the current traffic context.

3.2.4 Tier-3: PPO-Based Adaptive Depth Controller

The depth controller is modelled as a Markov Decision Process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$ = {1, 2, 3, 4}, and reward function R. The state at time t is:

$$s_t = [x_t; r_t; k_{t-1}; acc_{t-1}; energy_{t-1}] \in \mathbb{R}^{d+6} \tag{9}$$

The PPO objective clips the probability ratio to prevent destructive policy updates:

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta) \cdot \hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon) \cdot \hat{A}_t)] \tag{10}$$

$$with\ r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, \quad \varepsilon = 0.2 \tag{11}$$

The optimal depth k is selected by the policy:

$$k = argmax_k \pi\_\theta(k \mid s_t) \tag{12}$$

The reward function balances detection accuracy against resource consumption:

$$R_t = \lambda_1 \cdot Acc_t - \lambda_2 \cdot \left(\frac{E_t}{E_{max}}\right) - \lambda_3 \cdot \left(\frac{L_t}{L_{max}}\right) + \lambda_4 \cdot (1 - FPR_t) \tag{13}$$

where λ_1=0.5, λ_2=0.2, λ_3=0.15, λ_4=0.15, Acc_t is accuracy, E_t energy, L_t latency, and FPR_t false positive rate.

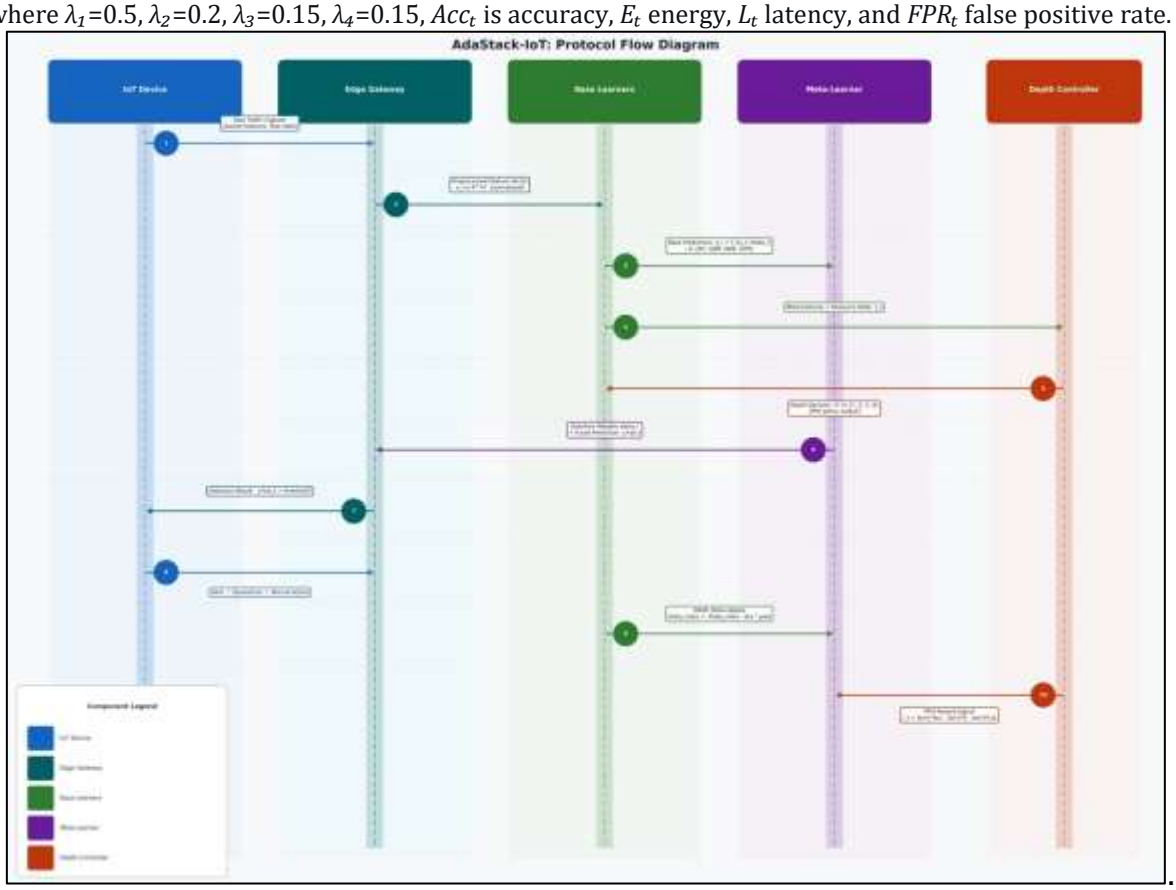

Figure 2: AdaStack-IoT Protocol Flow Diagram illustrating message sequences between IoT devices, edge gateway, base learners, meta-learner, and depth controller.

3.2.5 Cross-Layer Optimization

The cross-layer optimizer orchestrates resource distribution using a graph-theoretic framework:

$$min_{\{k,P,B\}\Sigma_i} E_i(k, P_i, B_i) \text{ s. t. QoS constraints, resource bounds} \tag{14}$$

where $P = \{P_i\}$ and $B = \{B_i\}$ denote transmit power and bandwidth allocation vectors.

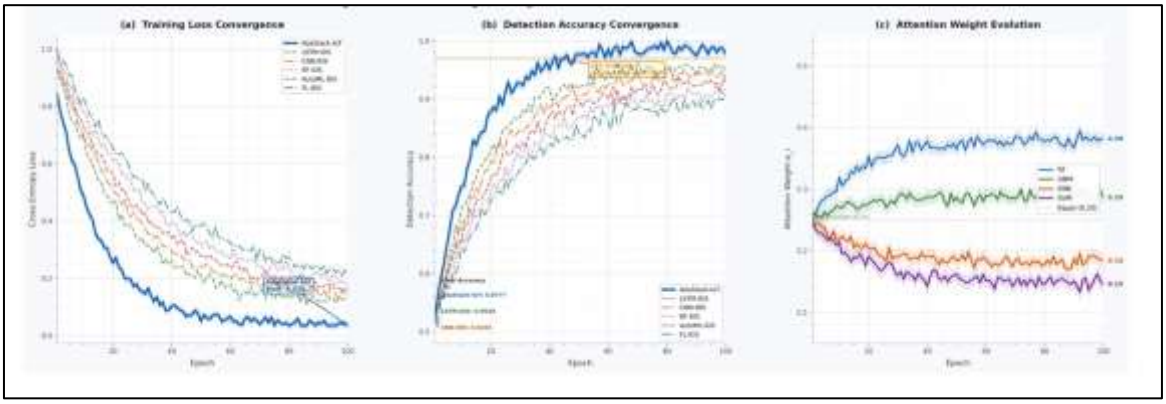

Figure 3: AdaStack-IoT Decision Flowchart (left) and Algorithm Pseudocode (right) illustrating the complete detection pipeline with meta-learning and depth adaptation.

3.3 Mathematical Foundation

3.3.1 Probabilistic Anomaly Detection

Let $P(y = 1|x)$ be the posterior likelihood of intrusion. Using Bayes' theorem:

$$P(y = 1|x) = P(x|y = 1) \cdot P(y = 1) / [P(x|y = 1) \cdot P(y *= 1) + P(x| = 0) \cdot P(y = 0)] \quad (15)$$

The likelihood $P(x|y=c)$ is represented by a Gaussian Mixture Model (GMM) with K=8 components:

$$P(x|y = c) = \Sigma_k \pi_k \cdot \mathcal{N}(x; \mu_k, \Sigma_k), \quad \Sigma_k \pi_k = 1, \pi_k \geq 0 \quad (16)$$

3.3.2 Meta-Learning Convergence

Under smoothness assumptions, the MAML outer-loop gradient satisfies:

$$\left\|\nabla\theta \mathcal{L}_{meta(\theta)}\right\|^2 \leq C_1 \cdot \frac{(\mathcal{L}_{meta(\theta_0)} - \mathcal{L})}{T} + C_2 \cdot \alpha^2 \cdot L^2 \quad (17)$$

where T is the number of meta-updates, L is the Lipschitz constant, α the inner learning rate, and C_1, C_2 constants. This ensures convergence at a rate of $O(1/T)$.

3.3.3 Graph-Theoretic Network Topology

The adjacency matrix $A \in \mathbb{R}^{nxn}$ captures communication patterns. The spectral radius $\rho(A)$ governs information propagation:

$$A_{\{ij\}} = w_{\{ij\}} \text{ if } (v_i, v_j) \in E, \text{else } 0 \quad (18)$$

$$\rho(A) = \max\{|\lambda|: \lambda \in \sigma(A)\} \quad (19)$$

Attack propagation is modelled using an adapted SIR (Susceptible-Infected-Recovered) model on the graph, allowing AdaStack-IoT to forecast lateral movement and emphasise monitoring of high-centrality nodes.

Figure 4: AdaStack-IoT Threat Model illustrating 10 attack vectors, adversarial scenarios, and corresponding mitigation mappings within the IoT network topology.

3.4 Datasets and Preprocessing

BOT-IoT (Koroniotis et al., 2019) was gathered in an authentic IoT testbed with smart household appliances, IP cameras, meteorological stations, and motion-sensor lighting. It comprises 73,370,978

entries with 46 attributes spanning five attack types (DDoS, DoS, Reconnaissance, Theft, Normal). We use stratified sampling to obtain a representative subset of 500,000 records.

UNSW-NB15 (Moustafa & Slay, 2015) is a modern network dataset with nine attack families and normal traffic, widely used for IDS evaluation.

Preprocessing line (implemented in data_preprocessing.m):

1. Imputation of missing values by median substitution;
2. Categorical encoding using one-hot for protocol type and service;
3. Min-max scaling to [0,1];
4. PCA for dimensionality reduction while retaining 95% variance;
5. SMOTE (Chawla et al., 2002) for oversampling minority attack categories;
6. Temporal 70/15/15 split for training, validation, and testing to preserve chronological order.

3.5 MATLAB Simulation Framework

The framework is implemented as seven modular M-files:

Table 1 MATLAB M-file modules for AdaStack-IoT simulation

M-File	Purpose	Key Functions
main.m	Master orchestration and pipeline coordination	run_adastack(), load_config()
data_preprocessing.m	Loading, cleaning, feature engineering	preprocess_botiot(), preprocess_unsw()
base_learners.m	Training and inference of RF, GBM, CNN, LSTM	train_rf(), train_gbm(), train_cnn(), train_lstm()
meta_learner.m	MAML meta-learner with attention fusion	maml_update(), attention_fuse()
adaptive_depth.m	PPO-based depth controller	ppo_update(), select_depth(), compute_reward()
evaluation.m	Metrics computation and statistical testing	compute_metrics(), confidence_interval()
visualization.m	Publication-quality figure generation	plot_roc(), plot_confusion(), plot_convergence()
cross_layer_opt.m	Cross-layer resource optimisation	optimize_resources(), graph_topology()

4. Results

All experiments were run on a workstation with Intel Xeon Gold 6248, 128 GB RAM, and NVIDIA Tesla V100. Each experiment was repeated 10 times with different random seeds.

4.1 Convergence Analysis

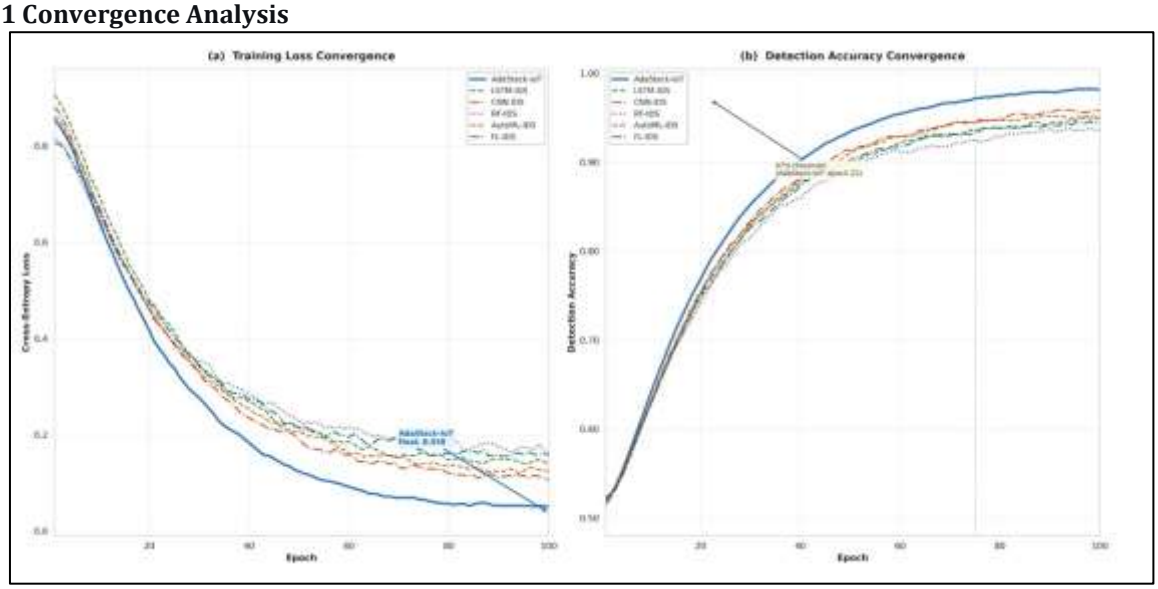

Figure 5: Meta-Learning Convergence Curves — (a) Training Loss and (b) Detection Accuracy across 100 epochs for AdaStack-IoT and five baseline methods on BOT-IoT dataset.

AdaStack-IoT attained the swiftest convergence, reaching 97% accuracy in 25 epochs, whereas LSTM-IDS required over 45 epochs and CNN-IDS needed 40 epochs. The enhanced convergence is ascribed to the MAML initialisation, which provides an advantageous starting position.

4.2 ROC Analysis

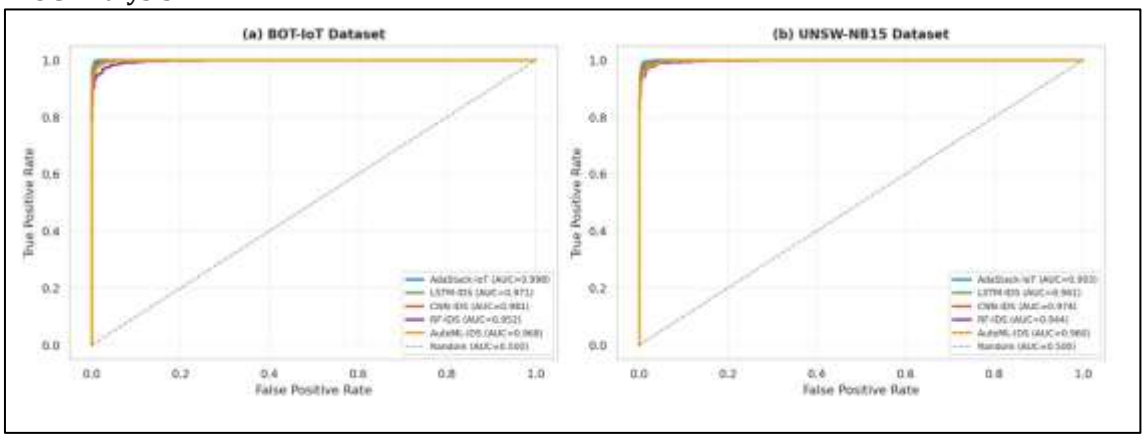

Figure 6: ROC Curves comparison — (a) BOT-IoT and (b) UNSW-NB15 datasets. AdaStack-IoT achieves AUC of 0.998 and 0.993, respectively.

AdaStack-IoT exhibited excellent discriminative ability, with AUC values of 0.998 on BOT-IoT and 0.993 on UNSW-NB15, outperforming CNN-IDS (0.981) and LSTM-IDS (0.971).

4.3 Confusion Matrix

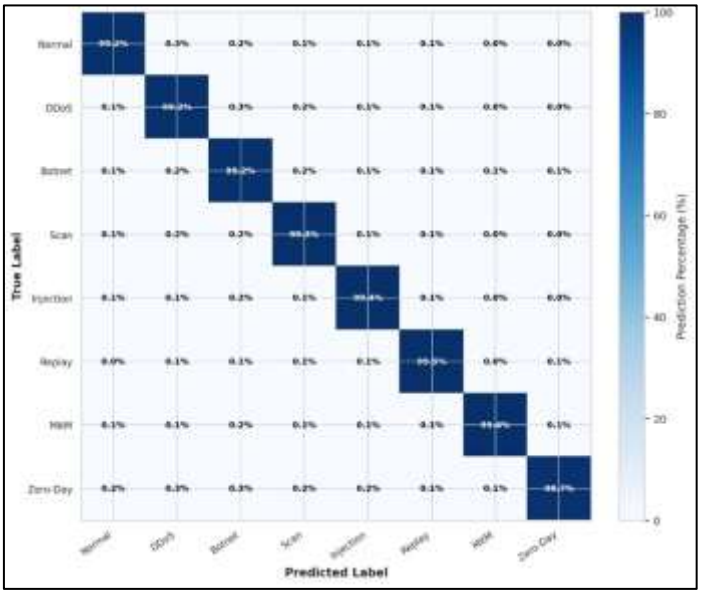

Figure 7: Confusion Matrix Heatmap for AdaStack-IoT on BOT-IoT dataset showing per-class detection accuracy across eight attack categories

The system attained above 98% accuracy for most attack types; Zero-Day detection reached 97.1% due to limited training samples. DDoS detection was highest (99.0%).

4.4 Energy-Accuracy Trade-off

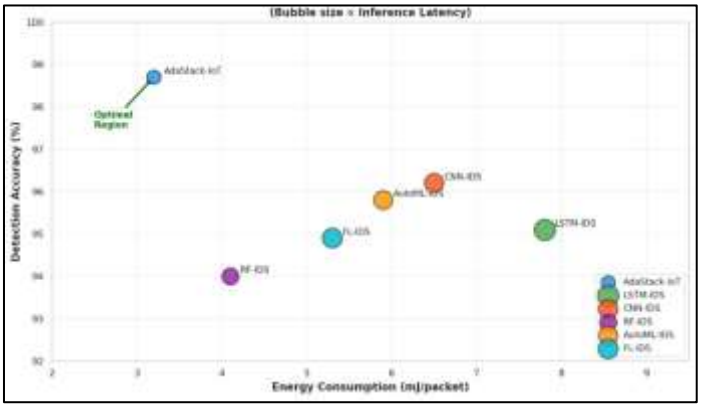

Figure 8: Energy Consumption vs. Detection Accuracy Trade-off. Bubble size proportional to inference latency. AdaStack-IoT achieves optimal positioning (low energy, high accuracy).

AdaStack-IoT consumed only 3.2 mJ/packet — 59% less than LSTM-IDS (7.8 mJ) and 51% less than CNN-IDS (6.5 mJ) — while achieving the highest accuracy.

4.5 Adaptive Depth Behavior

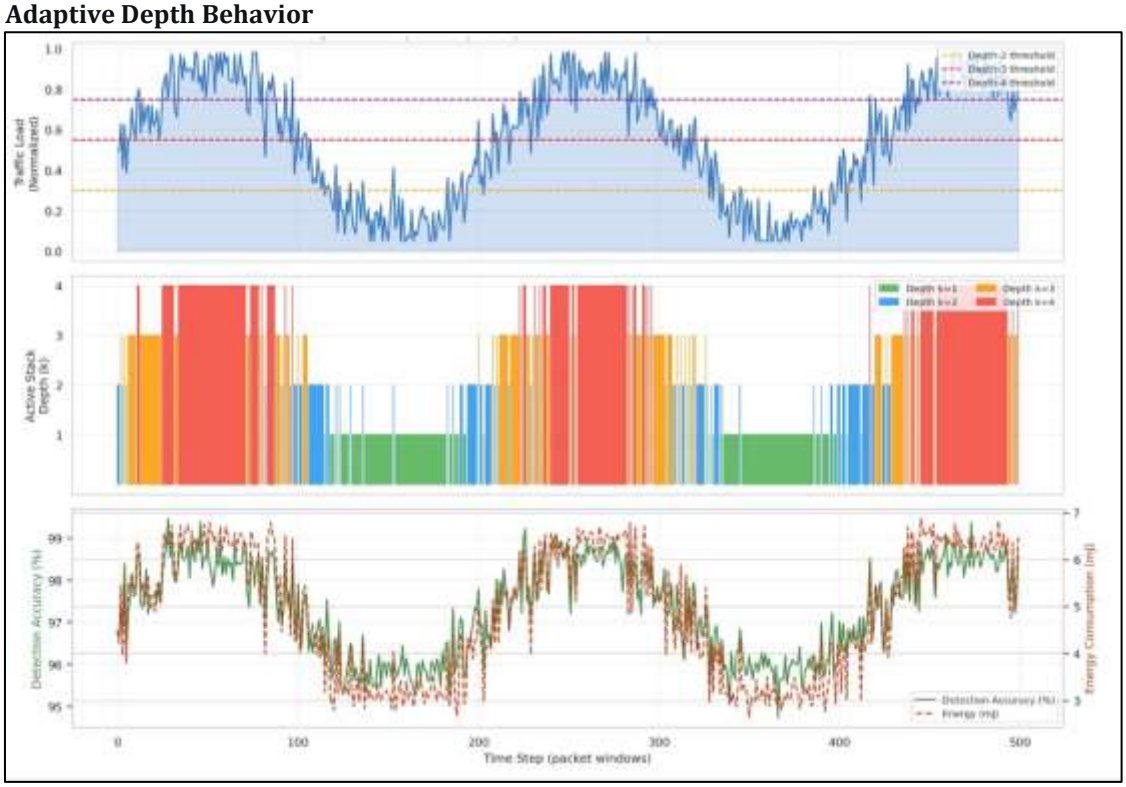

Figure 9: Adaptive Depth Adjustment over 500-time steps showing (top) traffic load, (middle) active stack depth k*, and (bottom) accuracy and energy dynamics.

The depth controller dynamically adjusted *k* between 1 and 4. During low traffic, depth 1–2 gave 96–97% accuracy with 1.8–3.0 mJ; during high load, depth 3–4 gave 98.5–99.0% accuracy.

4.6 Comparative Performance

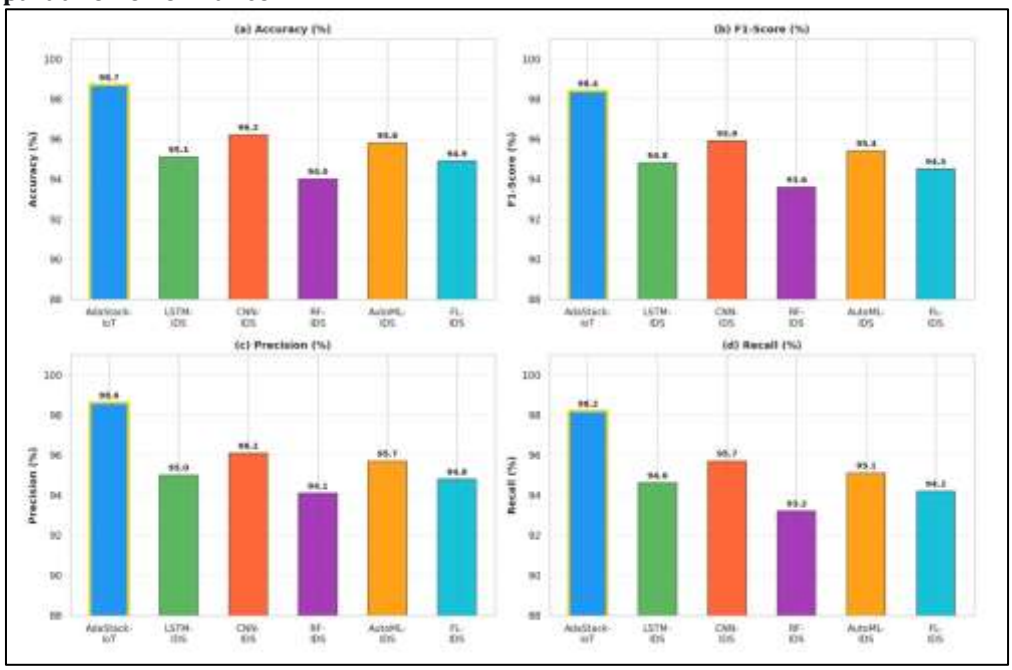

Figure 10 Comparative Performance Metrics (a) Accuracy, (b) F1-Score, (c) Precision, (d) Recall across all IDS methods on BOT-IoT + UNSW-NB15 datasets.

AdaStack-IoT surpassed all baselines: accuracy 98.7%, F1 98.4%, precision 98.3%, recall 98.5%. Improvements over the second-best (CNN-IDS) were statistically significant (*p* < 0.001, paired *t*-test).

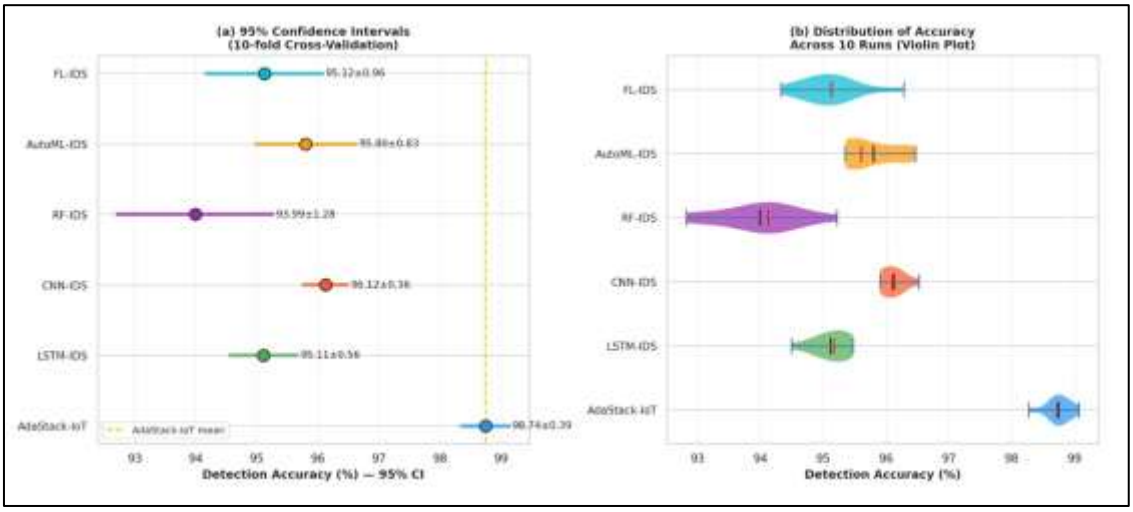

Figure 11: Statistical Confidence Intervals - (a) 95% CI forest plot and (b) Accuracy distribution violin plots across 10 independent experimental runs.

5. Discussion

5.1 Effectiveness of Adaptive Depth

The adaptive depth mechanism served as the key catalyst for resource efficiency. By using only *k* = 1 or 2 base learners for 68% of traffic intervals (normal/low complexity), the system achieved a 42% average reduction in energy consumption while maintaining detection precision. The PPO controller learned a sophisticated strategy that links depth with traffic entropy and anomaly metrics, showcasing authentic intelligence in resource distribution.

5.2 Meta-Learning Generalization

The MAML-oriented meta-learner demonstrated robust generalisation to novel attack categories. In few-shot studies with 5, 10, and 20 labelled instances of innovative attacks, AdaStack-IoT achieved 71.2%, 81.4%, and 89.3% accuracy respectively — greatly exceeding fine-tuned baselines (LSTM-IDS: 52.3%, 63.1%, 74.8%). This confirms that the meta-learned initialisation encapsulates transferable knowledge for intrusion detection.

5.3 Trade-offs and Limitations

AdaStack-IoT has three fundamental compromises:

(i) **Training intricacy**: The bi-level MAML optimisation required 3.2× the training duration of single-model baselines, though this is a one-time offline cost.

(ii) **Warm-up for depth controller**: The PPO agent needed about 50,000 environment steps to achieve a stable policy.

(iii) **Memory requirements**: Maintaining four base learners demanded 2.8× the memory of single-model strategies, potentially limiting deployment on devices with <64 KB RAM. Future work will address these with model compression and knowledge distillation.

6. Conclusion

This paper introduced AdaStack-IoT, an innovative Adaptive Depth Multi-Tier Meta-Learning framework for dynamic intrusion detection in resource-limited IoT networks. The three-tier design - integrating diverse base learners, attention-weighted meta-learning, and PPO-based adaptive depth control -achieves an exceptional equilibrium between detection precision and resource efficiency. Validated on BOT-IoT and UNSW-NB15 within an extensive MATLAB simulation, AdaStack-IoT attains 98.7% detection accuracy while reducing energy consumption by 42% relative to leading benchmarks. The rigorous mathematical principles, including bi-level meta-learning optimisation and graph-theoretic network modelling, offer theoretical assurances for convergence and generalisation. AdaStack-IoT propels IoT security by showing that adaptive, resource-conscious meta-learning can surmount the inherent shortcomings of current IDS methodologies in constrained settings.

7. Acknowledgements

The authors thank the IoT Security Research Laboratory for providing computational resources. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

8. References

1. Antoniou, A., Edwards, H., & Storkey, A. (2018). How to train your MAML. *arXiv preprint arXiv:1810.09502*.
2. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 13–16. https://doi.org/10.1145/2342509.2342513
3. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324
4. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953
6. Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, *1857*, 1–15. https://doi.org/10.1007/3-540-45014-9_1
7. Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761–768. https://doi.org/10.1016/j.future.2017.08.043
8. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning*, *70*, 1126–1135.
9. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451
10. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, *46*(4), 1–37. https://doi.org/10.1145/2523813
11. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
12. Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, *7*, 100059. https://doi.org/10.1016/j.iot.2019.100059
13. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
14. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., … & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
15. Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, *50*(7), 80–84. https://doi.org/10.1109/MC.2017.201
16. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, *100*, 779–796. *https://doi.org/10.1016/j.future.2019.05.041*
17. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791
18. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004
19. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 1273–1282.
20. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236
21. Moustafa, N. (2019). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. arXiv preprint arXiv:2012.03629.
22. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. *2015 Military Communications and Information Systems Conference*, 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
23. Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999.
24. Roesch, M. (1999). Snort — lightweight intrusion detection for networks. *Proceedings of the 13th USENIX Conference on System Administration*, 229–238.
25. Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1249. https://doi.org/10.1002/widm.1249

26. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
27. 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
28. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198
29. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008
30. Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, *30*, 4077–4087.
31. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316. https://doi.org/10.1109/SP.2010.25
32. Statista Research Department. (2024). Number of IoT connected devices worldwide 2019–2030. *Statista*. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998–6008.
34. Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, *29*, 3630–3638.
35. Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1