UDC 004.056.5

DETECTION OF MALICIOUS ANOMALIES IN IOT-DEVICES USING DEEP LEARNING

Kosheleva E.D.
master's degree, Kuban state technological university
(Krasnodar, Russia)

Klychev A.M.
master's degree, Kuban state technological university
(Krasnodar, Russia)

ОБНАРУЖЕНИЕ ВРЕДОНОСНЫХ АНОМАЛИЙ В IOT-УСТРОЙСТВАХ С ПОМОЩЬЮ ГЛУБИННОГО ОБУЧЕНИЯ

Кошелева Э.Д.
магистр, Кубанский государственный технологический
университет (Краснодар, Россия)

Клычев А.М.
магистр, Кубанский государственный технологический
университет (Краснодар, Россия)

Abstract

Deep learning techniques are increasingly used to detect malicious anomalies in IoT environments, where traditional security mechanisms fail to scale or adapt to dynamic data flows. This study evaluates various neural network architectures suitable for constrained devices, analyzes deployment models from edge to cloud, and examines the resilience of DL systems to adversarial threats. Practical implementation details, including preprocessing pipelines and lightweight inference, are presented alongside comparative performance metrics. Challenges related to data availability, explainability, and system heterogeneity are identified as critical barriers to widespread adoption.

Keywords: IoT security, anomaly detection, deep learning, edge computing, neural networks, adversarial robustness, model deployment, cyber threats.

Аннотация

Глубокие нейросетевые методы находят всё более широкое применение для обнаружения вредоносных аномалий в среде IoT, где традиционные средства защиты оказываются недостаточными. Представлены различные архитектуры нейронных сетей, адаптированные к ресурсным ограничениям устройств, рассмотрены стратегии развёртывания моделей от локального уровня до облака и проведён анализ устойчивости к атакующим воздействиям. Описаны подходы к предобработке данных и организации облегчённого инференса. Особое внимание уделено проблемам доступности данных, объяснимости моделей и разнообразия аппаратных платформ.

Ключевые слова: безопасность IoT, обнаружение аномалий, глубокое обучение, edge-вычисления, нейронные сети, устойчивость к атакам, развёртывание моделей, киберугрозы.

Introduction

The proliferation of Internet of Things (IoT) devices has redefined the architecture of modern digital ecosystems. These devices, deployed across industrial, medical, and consumer domains, generate continuous streams of sensor data and operational telemetry. Despite their utility, IoT environments are inherently vulnerable to security threats due to limited computational resources, weak authentication schemes, and diverse firmware configurations. Consequently, the detection of malicious anomalies within such systems remains a critical challenge in maintaining the integrity of connected infrastructures.

Traditional rule-based security mechanisms, while effective in constrained scenarios, lack adaptability and scalability when applied to dynamic and heterogeneous IoT networks. Signature-based methods often fail to detect novel or obfuscated attacks, and their performance deteriorates as data volume increases. In contrast, deep learning (DL) techniques offer promising alternatives by enabling the automatic extraction of abstract features and pattern recognition from raw data. When trained on representative datasets, DL models can identify previously unseen malicious behaviors with high accuracy and minimal manual intervention.

The aim of this study is to investigate the applicability of DL-based architectures for detecting malicious anomalies in IoT environments. The research focuses on evaluating neural models for anomaly classification, analyzing deployment constraints specific to edge computing devices, and presenting visual and tabular comparisons of performance metrics. Through this work, we seek to outline the practical considerations and technical advantages associated with using intelligent detection systems in IoT-based infrastructures.

Main part. IoT architecture and threat landscape

The architecture of IoT ecosystems is inherently decentralized, composed of interconnected edge devices, local gateways, and cloud services. Devices often operate with minimal supervision, limited firmware protection, and weak cryptographic modules. These factors, combined with large-scale deployments and wireless communication channels, make IoT infrastructures attractive targets for malicious actors. Attacks may originate from compromised firmware, lateral movement within networks, or spoofed control commands that exploit weak authentication protocols.

Common threat vectors include botnet traffic propagation, firmware injection, distributed denial-of-service (DDoS) attempts, port scanning, and brute-force access to credentials. These anomalies may not immediately disrupt functionality but can significantly compromise data integrity, network availability, and system trustworthiness. Effective detection requires monitoring both network traffic and device-level behavioral signatures [1].

As shown in figure 1, botnet-related anomalies account for the largest portion of malicious activity detected in IoT environments, followed by firmware injection and denial-of-service patterns. This distribution highlights the need for anomaly detection systems that can differentiate subtle variations in traffic patterns and behavioral deviations.

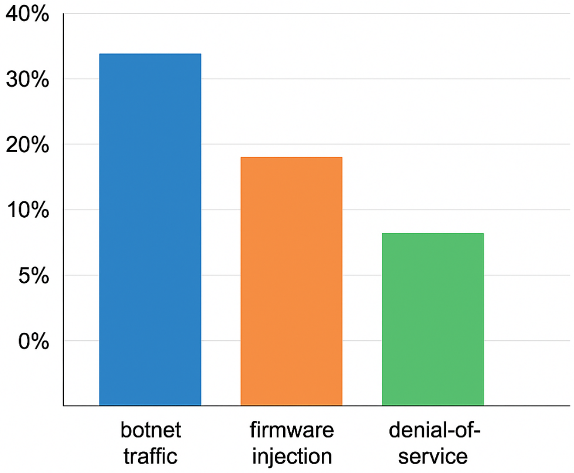

Figure 1. Prevalence of malicious activity types in IoT environments

The figure illustrates the relative frequency of major attack types targeting IoT devices based on recent empirical datasets. Botnet traffic is the most common, reflecting the ease with which devices can be recruited into large-scale coordinated attacks, while firmware injection remains a critical concern due to its persistence and stealth.

Deep learning models for anomaly detection in iot systems

The application of DL methods in the detection of malicious anomalies within IoT systems requires careful alignment between algorithmic complexity and resource constraints. Unlike traditional enterprise networks, IoT devices often operate under limited computational capacity, power restrictions, and connectivity variability. As a result, models deployed in such contexts must balance detection accuracy with efficiency and portability.

Convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and autoencoders are among the most widely adopted DL architectures for anomaly detection in streaming data. CNNs are effective for extracting spatial patterns from preprocessed network traffic features, while LSTMs capture temporal dependencies in sequential telemetry data. Autoencoders, especially in their variational form, can learn compact representations of normal behavior and identify deviations as reconstruction errors [2].

Edge deployment imposes additional constraints: inference latency must remain low, model size must be minimal, and updates must be incrementally applied without complete retraining. To address these issues, model compression techniques such as pruning, quantization, and knowledge distillation are frequently applied after training. Moreover, training itself is typically conducted offline in centralized environments, and only the final inference model is exported to the device.

The example below illustrates a compact feedforward neural network in PyTorch, designed for binary anomaly classification. The model accepts engineered feature vectors derived from packet metadata or system logs and outputs a probability of malicious behavior.

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class IoTAnomalyDetector(nn.Module):
    def __init__(self, input_dim):
        super(IoTAnomalyDetector, self).__init__()
        self.fc1 = nn.Linear(input_dim, 64)
        self.fc2 = nn.Linear(64, 32)
        self.dropout = nn.Dropout(0.25)
        self.output = nn.Linear(32, 2)  # 2 classes: normal, anomaly

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = self.dropout(F.relu(self.fc2(x)))
        return self.output(x)
```

This architecture is intentionally shallow and lightweight, making it suitable for execution on edge hardware such as ARM-based microcontrollers or embedded Linux boards. While more complex architectures may yield marginal improvements in accuracy, they do so at the cost of inference speed and energy consumption-two critical parameters in real-time IoT systems [3].

Training such models requires labeled datasets that capture both benign and malicious activity, with an emphasis on generalizability across device types and usage scenarios. Data augmentation and regularization techniques help prevent overfitting, especially when anomaly samples are scarce or imbalanced. Ultimately, the success of DL-based anomaly detection in IoT hinges on its ability to adaptively generalize while remaining lightweight and interpretable.

Model comparison for anomaly detection in IoT environments

The choice of deep learning architecture for anomaly detection in IoT systems must be driven by a balance between classification performance, resource efficiency, and adaptability to streaming conditions. Unlike general-purpose computing environments, IoT deployments operate under highly constrained conditions where even modest increases in model complexity can render real-time detection impractical. As a result, model selection must consider not only accuracy metrics but also inference latency, memory footprint, and robustness to noisy or incomplete data.

Three dominant classes of models are evaluated in this context: feedforward networks (FFNs), recurrent architectures such as LSTM networks, and autoencoder-based anomaly detection systems. FFNs are computationally lightweight and well-suited for tabular input derived from structured logs or metadata. LSTMs, on the other hand, offer improved performance in time-series contexts, where sequence dependencies are essential for capturing temporal anomalies. Autoencoders provide a flexible unsupervised approach, capable of detecting unknown attacks by modeling the normal operational space and identifying deviations as reconstruction errors.

Table 1 presents a comparative analysis of these models based on empirical benchmarks from IoT-specific datasets. The evaluation includes multiple criteria: detection accuracy, false positive rate (FPR), average inference time per sample, and model size (in MB) after compression. All models were trained on the same preprocessed dataset and evaluated using a standardized test protocol.

Table 1

Comparative evaluation of DL models for anomaly detection in IoT

Model	**Accuracy (%)**	**False positive rate (%)**	**Inference time (ms)**	**Model size (MB)**
Feedforward NN	91.2	5.4	3.2	0.7
LSTM	94.7	3.1	12.8	3.5
Autoencoder	89.6	6.2	5.6	1.1

The results show that LSTM networks achieve the highest overall accuracy (94.7%) and lowest FPR (3.1%), making them ideal for high-integrity anomaly detection in streaming telemetry. However, they exhibit longer inference times and larger model sizes, which may limit their feasibility on low-power devices. FFNs offer a compact and fast alternative, sacrificing a small margin in accuracy for a 4x reduction in latency. Autoencoders provide strong performance in unsupervised scenarios but require careful tuning to avoid underfitting normal patterns or overflagging benign anomalies.

Ultimately, the model choice should align with the deployment target: LSTMs are appropriate for centralized or gateway-based analysis, while FFNs and compressed autoencoders are better suited for edge-level detection on individual IoT devices. Hybrid strategies, in which lightweight models flag suspicious behavior and forward events to a centralized engine for deeper analysis, may offer an optimal balance between responsiveness and detection fidelity.

Deployment strategies for DL-based anomaly detection in IoT

The integration of deep learning models into IoT security infrastructure requires deployment strategies that align with the operational characteristics of the system. Unlike traditional IT environments, IoT networks often exhibit constraints in processing power, memory availability, connectivity bandwidth, and update frequency. Therefore, deployment scenarios must be carefully selected based on the location of inference, required response time, and data sensitivity.

Deployment on a single device (local edge) prioritizes minimal latency and independence from external connectivity. In this configuration, a small feedforward neural network is embedded directly into the device firmware or local runtime environment. This setup is optimal for scenarios requiring millisecond-level decisions, such as real-time access control or actuator response. However, it restricts model complexity and update frequency, making it suitable primarily for known threat profiles.

Gateway-level aggregation offers a trade-off between performance and visibility. Data from multiple devices is aggregated at a local node, where models such as LSTMs can be used to analyze temporal patterns and detect coordinated anomalies [4]. This architecture allows for more powerful detection while maintaining acceptable inference latency and supporting periodic model updates.

Cloud-based central analysis provides the highest analytical power by utilizing sophisticated architectures like autoencoders or transformers on aggregated data. This approach enables continuous model refinement and access to broader datasets, which improves detection quality. However, it introduces latency and requires reliable connectivity, which may not be suitable for latency-sensitive IoT applications.

Federated learning is an emerging approach that enables distributed training across multiple IoT nodes without sharing raw data. Lightweight models such as pruned FFNs or LSTMs are updated locally and only gradients or model updates are transmitted securely. This model preserves privacy, supports adaptability, and reduces communication overhead-but depends on synchronization and secure update aggregation.

A hybrid cascade detection strategy combines local lightweight models for initial anomaly scoring with selective forwarding of suspicious samples to a centralized analysis engine. This architecture balances response time and accuracy while minimizing data transfer. It is particularly effective in scenarios where bandwidth is limited but detection confidence must remain high.

Table 2 summarizes the comparative characteristics of these deployment strategies across multiple operational dimensions.

Table 2

Deployment strategies for DL-based anomaly detection in IoT

Deployment scenario	**Model type**	**Latency requirement**	**Connectivity dependence**	**Update strategy**
Single-device (local edge)	Feedforward NN	Ultra-low (<5 ms)	Low	Static model
Gateway-level aggregation	LSTM	Low to moderate (5–20 ms)	Medium	Periodic batch update
Cloud-based central analysis	Autoencoder / Transformer	Moderate to high (20–100 ms)	High	Continuous retraining
Federated learning (multi-device)	Compressed FFN or LSTM	Moderate (10–30 ms)	Medium	Secure federated update
Hybrid cascade detection	Lightweight local + full remote	Split latency (2–50 ms)	Variable	Trigger-based escalation

The table highlights the trade-offs associated with each strategy and illustrates how deployment decisions impact model selection, update mechanisms, and latency constraints. No single approach fits all scenarios; rather, deployments must be tailored to the specific constraints and objectives of the target IoT application domain.

Data preprocessing and lightweight inference on IoT edge devices

The effectiveness of DL-based anomaly detection in IoT settings depends not only on the architecture of the model but also on the preprocessing pipeline and the runtime execution strategy. In edge scenarios, both stages must be computationally efficient, modular, and capable of handling incomplete or noisy sensor input.

Preprocessing typically involves normalization, feature extraction, and dimensionality reduction. For structured IoT logs or packet-level telemetry, relevant features include source and destination IP entropy, payload size variability, time between packets, and protocol distribution. These features are computed using sliding windows and prepared in fixed-size vectors suitable for input into a compact neural network [5].

The example below shows a lightweight preprocessing and inference pipeline using PyTorch and NumPy, tailored for execution on embedded Python runtimes such as MicroPython or edge containers.

```
import numpy as np
import torch
import torch.nn.functional as F
from trained_model import IoTAnomalyDetector  # pretrained model class
```

```
# Example input: feature vector from IoT device log
def extract_features(packet_stream):
    avg_payload = np.mean([p['payload_size'] for p in packet_stream])
    std_interval = np.std([p['timestamp_diff'] for p in packet_stream])
    proto_count = len(set(p['protocol'] for p in packet_stream))
    return np.array([avg_payload, std_interval, proto_count], dtype=np.float32)

# Normalize and convert to tensor
def normalize_and_infer(feature_vector, model, mean, std):
    normalized = (feature_vector - mean) / std
    x_tensor = torch.tensor(normalized).unsqueeze(0)
    with torch.no_grad():
        logits = model(x_tensor)
        probabilities = F.softmax(logits, dim=1)
    return probabilities[0][1].item()  # probability of anomaly

# Sample execution
features = extract_features(packet_stream)
model = IoTAnomalyDetector(input_dim=3)
model.load_state_dict(torch.load("iot_detector.pt", map_location='cpu'))
model.eval()

anomaly_score = normalize_and_infer(features, model, mean=np.array([200, 0.5, 4]), std=np.array([100, 0.2, 2]))
print(f"Anomaly probability: {anomaly_score:.3f}")
```

This compact pipeline can be deployed in environments where full-stack frameworks like TensorFlow are too resource-intensive. Once trained and quantized, the model can be stored in less than 1 MB, enabling fast and reliable predictions directly on the device or local gateway. For deployments with intermittent connectivity, the output (anomaly score) can be logged locally or sent asynchronously to a central server only when it crosses a configurable threshold.

By combining efficient feature engineering with optimized inference routines, this setup enables real-time anomaly detection that is both robust and operationally viable across a range of embedded IoT platforms.

Resilience of DL models to adversarial threats in IoT systems

Despite their effectiveness in detecting complex anomalies, DL models deployed in IoT environments are not immune to adversarial threats. In fact, their reliance on learned representations makes them susceptible to subtle manipulations in the input space or model internals, especially when deployed in open or distributed settings. Understanding and mitigating these vulnerabilities is critical to preserving the reliability of anomaly detection systems.

One common attack vector is adversarial perturbation, in which an attacker slightly modifies legitimate inputs to induce misclassification without altering the data in a semantically noticeable way. Feedforward and recurrent models are particularly vulnerable to such perturbations, especially when trained without noise-aware regularization. These attacks can be countered using adversarial training or input smoothing techniques that improve the model's robustness.

Data poisoning attacks occur during the training phase and are especially relevant in federated or decentralized systems. By injecting carefully crafted samples into the training dataset, an attacker can degrade model performance or induce false negatives on malicious patterns. Defensive strategies include robust aggregation mechanisms, outlier detection, and selective validation of incoming data.

More subtle threats include model inversion, which exploits access to outputs or gradients to reconstruct sensitive input data. This attack is most feasible in centralized deployments of lightweight models. Encryption of model parameters and restricted inference access are the primary countermeasures [6].

In network-level scenarios, evasion via protocol mimicry is a growing concern. Attackers craft traffic patterns that imitate benign behavior to avoid detection by models trained on protocol statistics or packet structure. Autoencoders are particularly at risk here, as they often fail to distinguish structurally valid but semantically harmful input. Solutions involve integrating protocol-specific fingerprinting and traffic segmentation into the feature engineering pipeline.

Finally, gradient leakage poses a privacy risk in federated settings. If updates from client devices are not aggregated securely, an attacker can infer training data characteristics from shared gradients. This risk is mitigated through secure aggregation protocols and differential privacy techniques that introduce randomness into updates without significantly compromising learning quality [7].

Table 3 summarizes key attack types, their impact on DL-based IoT detection systems, and corresponding mitigation strategies.

Table 3

Resilience of DL models to adversarial threats in IoT systems

Attack type	Vulnerable models	Impact	Mitigation strategies
Adversarial perturbation	FFN, LSTM	Misclassification of malicious input	Adversarial training, input smoothing
Data poisoning	All (training-stage issue)	Model degradation and false negatives	Robust training, data validation
Model inversion	FFN (centralized deployment)	Exposure of input data patterns	Access control, model encryption
Evasion via protocol mimicry	Autoencoder	Undetected malicious traffic patterns	Traffic fingerprinting, statistical modeling
Gradient leakage	Federated LSTM	Loss of privacy in local updates	Secure aggregation, differential privacy

As the threat landscape evolves, designing resilient learning architectures must become a core consideration in IoT security. This includes not only protecting against direct attacks but also enabling post-deployment model auditing and anomaly explanation to improve transparency and trust.

Practical limitations of DL-based anomaly detection in IoT

While deep learning has demonstrated promising results in detecting malicious activity within IoT infrastructures, its real-world adoption faces several practical constraints. These limitations span technological, organizational, legal, and operational domains, each of which must be addressed to facilitate secure and sustainable deployment [8].

One of the primary challenges is the scarcity of high-quality labeled datasets. Effective supervised training requires large volumes of diverse anomaly-labeled data, which is rarely available in IoT deployments. Many organizations lack centralized logging infrastructure or data retention policies, leading to fragmented, incomplete, or non-standardized datasets. Furthermore, privacy regulations may restrict the collection of raw telemetry data, especially in medical or consumer-facing applications.

Another obstacle is the fragmentation of IoT device ecosystems. Vendors utilize different hardware platforms, operating systems, and communication protocols, resulting in highly heterogeneous environments [9]. DL models trained on one device type or network context may not generalize well to others without additional tuning or retraining. This hampers model portability and increases the cost of cross-platform support.

Regulatory and compliance considerations also play a critical role. For example, anomaly detection models deployed in regulated industries (e.g., healthcare, critical infrastructure) must be explainable and certifiable [10]. Many DL systems operate as "black boxes," making it difficult to justify security decisions to auditors or regulatory authorities. Efforts to integrate explainable AI into IoT security remain limited and experimental.

In terms of operational maintenance, updating DL models in the field is a non-trivial process. Over-the-air updates may pose security risks themselves, while manual update cycles are labor-intensive and error-prone. Moreover, most IoT devices lack the capability for full retraining or real-time adaptation, requiring the use of transfer learning, distillation, or cloud-assisted update architectures.

Lastly, cost and energy consumption remain persistent concerns. Even with model compression and inference optimization, the deployment of DL components increases hardware complexity, power requirements, and overall system cost-particularly in battery-operated or large-scale sensor deployments.

Addressing these limitations will require advances not only in model architecture but also in system design, regulatory frameworks, data infrastructure, and vendor collaboration [11]. Until these gaps are resolved, the adoption of DL-based anomaly detection in IoT environments will likely be confined to controlled or high-value scenarios.

Conclusion

The increasing ubiquity of IoT devices across critical sectors has intensified the need for effective and scalable anomaly detection systems capable of identifying malicious behavior in real time. Traditional security mechanisms are ill-suited to the dynamic and resource-constrained nature of IoT environments, prompting a shift toward deep learning-based approaches that can learn complex behavioral patterns and generalize to previously unseen threats.

This study explored the architectural and operational dimensions of applying deep learning to anomaly detection in IoT networks. It examined key model types, deployment strategies, and their respective trade-offs in terms of accuracy, latency, and computational overhead. Furthermore, the work highlighted practical challenges including adversarial vulnerability, privacy risks, and the limited availability of labeled datasets for supervised learning. Through comparative analysis and code-level examples, the paper demonstrated how lightweight neural architectures can be effectively adapted to the IoT context while maintaining sufficient robustness and interpretability.

Although deep learning offers significant potential for enhancing IoT security, its successful deployment requires a multidimensional approach that considers infrastructure heterogeneity, regulatory compliance, update logistics, and adversarial resilience. Future research should focus on building adaptive, explainable, and energy-efficient models capable of long-term operation in real-world edge environments. In doing so, DL-based anomaly detection can become a foundational component of next-generation IoT cybersecurity frameworks.

References

1. Abusitta A., de Carvalho G.H., Wahab O.A., Halabi T., Fung B.C., Al Mamoori S. Deep learning-enabled anomaly detection for IoT systems // Internet of Things. 2023. Vol. 21. P. 100656.
2. Riaz S., Latif S., Usman S.M., Ullah S.S., Algarni A.D., Yasin A., Hussain S. Malware detection in internet of things (IoT) devices using deep learning // Sensors. 2022. Vol. 22. No. 23. P. 9305.
3. Diro A., Chilamkurti N., Nguyen V.D., Heyne W. A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms // Sensors. 2021. Vol. 21. No. 24. P. 8320.
4. Ullah I., Mahmoud Q.H. Design and development of a deep learning-based model for anomaly detection in IoT networks // IEEe Access. 2021. Vol. 9. P. 103906-103926.
5. Aversano L., Bernardi M.L., Cimitile M., Pecori R., Veltri L. Effective anomaly detection using deep learning in IoT systems // Wireless Communications and Mobile Computing. 2021. Vol. 2021. No. 1. P. 9054336.
6. Ahmad Z., Shahid Khan A., Nisar K., Haider I., Hassan R., Haque M.R., Rodrigues J.J. Anomaly detection using deep neural network for IoT architecture // Applied Sciences. 2021. Vol. 11. No. 15. P. 7050.

7. Dudak A., Israfilov A. Application of blockchain in IT infrastructure management: new opportunities for security assurance // German International Journal of Modern Science. 2024. № 92. P. 103-107.
8. Vigoya L., Fernandez D., Carneiro V., Novoa F.J. IoT dataset validation using machine learning techniques for traffic anomaly detection // Electronics. 2021. Vol. 10. No. 22. P. 2857.
9. Mukherjee I., Sahu N.K., Sahana S.K. Simulation and modeling for anomaly detection in IoT network using machine learning // International journal of wireless information networks. 2023. Vol. 30. No. 2. P. 173-189.
10. Jaramillo-Alcazar A., Govea J., Villegas-Ch W. Anomaly detection in a smart industrial machinery plant using iot and machine learning // Sensors. 2023. Vol. 23. No. 19. P. 8286.
11. Ali S., Abusabha O., Ali F., Imran M., Abuhmed T. Effective multitask deep learning for iot malware detection and identification using behavioral traffic analysis // IEEE Transactions on Network and Service Management. 2022. Vol. 20. No. 2. P. 1199-1209.