JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)

A PUBLICATION OF ASSOCIATION OF SCINCE EDUCATORS ANAMBRA STATE (ASEA)

Vol. 1(1); 2025

PRINT ISSN: 3092-9180 E-ISSN: 3115-4379

Editor in-Chief
PROF. JOSEPHINE N. OKOLI

JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)

Editor in-Chief
PROF. JOSEPHINE N. OKOLI

A PUBLICATION OF ASSOCIATION OF SCIENCE EDUCATORS ANAMBRA (ASEA)

JOURNAL OF *INNOVATIONS IN SCIENCE EDUCATION (JISE)*

Vol. 1 (1); 2025

JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)

Vol. 1(1); 2025

Print ISSN: 3092-9180
E- ISSN: 3115-4379

Published in July, 2025.

Printed in Nigeria in the year 2025 by:
Love Isaac Consultancy Services
No 1 Etolue Street, Ifite Awka, Anambra State, Nigeria
+234-803-549-6787, +234-803-757-7391

JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)

Vol. 1(1); 2025

EDITORIAL BOARD

JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)

Vol. 1(1); 2025

EDITORIAL

Journal of Innovations in Science Education (JISE) is a Publication of Association of Science Educators Anambra (ASEA). It is publishable both online and offline. The publication is twice a year. It embraces only on science education and innovative ideas. JIES provide an avenue for dissemination of research findings, innovative ideas and practices between researchers, science educators and policy makers in the form of original research, book review, theoretical and conceptual papers which will serve as an important reference for the advancement of teaching, learning and research in the field of science education.

We are grateful to the contributors and hope that our readers will enjoy reading these contributions.

Prof. Josephine N. Okoli
Editor-in-Chief

JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)

Vol. 1(1); 2025

TABLE OF CONTENTS

JOURNAL OF INNOVATIONS IN SCIENCE EDUCATION (JISE)
Vol. 1(1); 2025

ENERGY CONSUMPTION-BASED RANSOMWARE DETECTION ON ANDROID DEVICES

[1]Iyamah B. E, [2]Malasowe B. O., [3]Ojie D. V., [4]Monye S. N.
[1]boyce.iyamah@unidel.edu.ng, [2]malasowe.brigit@unidel.edu.ng,
[3]deborah.ojie@unidel.edu.ng, [4]ngozi.monye@unidel.edu.ng
[1 & 3] Department of Software Engineering, [2]Department of Computer Science, [4]Department of Information & Communication Technology

Abstract

An Internet of Things (IoT) architecture comprises heterogeneous sensors and actuators communicating over networked platforms to perform automated tasks.The purpose of this study is to quantify energy consumption in IoT deployments and identify key optimization strategies. We employ a quantitative experimental design across 100 simulated IoT nodes under varied load and scheduling conditions. Our analysis reveals that (1) adaptive load balancing reduces per-device power draw by 22 %, (2) firmware scheduling optimizations improve system efficiency by 14 %, and (3) real-time monitoring with a predictive algorithm anticipates consumption spikes with 88 % accuracy. We therefore recommend integrating these strategies into standard IoT frameworks to enhance energy efficiency and prolong device lifespan.

Keywords*:* Energy Consumption-Based, Ransomeware Detection, Internet of Things (IoT)

Introduction

It is anticipated by IBM that the number of Internet-connected devices will surpass the number of people, and that connectivity will continue to advance, reaching approximately 50 billion devices by 2030. The term Internet of Things (IoT) refers to the proliferation of connected objects in an actuation network. (Gubbi et al., 2021) a platform where sensors and actuators are integrated with the surroundings to exchange data and create a shared operating image. It is evident that the Internet of Things (IoT) is expanding globally. In 2016, the Dyn breach revealed the key weaknesses in smart networks. One of the most important concerns nowadays is Internet of Things (IoT) security.

The threat posed by infected Internet-connected objects not only compromises IoT security but also puts the entire Internet ecosystem at risk, as it may be able to take advantage of the weak points in the Things (smart gadgets) that have been set up as botnets. Via distributed denial of service (DDoS) assaults, Mirai malware crippled the Internet and compromised the video surveillance equipment. The complexity and diversity of security attack vectors have changed in recent years. Therefore, it is crucial to examine methods in the context of the Internet of Things in order to recognize, stop, or detect new assaults. By analyzing current protection strategies, this survey categorizes IoT security threats and problems for IoT networks.

A unique global identifier that is globally addressable at the start of an Internet of Things system is used to identify a single object. The information obtained from the object's access in this case can be as little as the static information stored on the RFID tags. As a result, items that are uniquely identified, linked to the Internet, and accessible (interactively) by other objects—referred to as "things" in this context—are considered to be part of the Internet of Things (IoT). The next major development that will transform the Internet into a fully integrated future Internet (of things) is the Internet of Things, or IoT.

This trend is being driven by the recent explosion in the adoption and integration of wireless network technology. In their 2013 study, Karimi and Atkinson claimed that expanding communication networks to include physical objects would accelerate the number of devices connected to the network and the amount of data that can be shared via the Internet. IoT provides ubiquitous connection for a wide range of devices, services, and applications. These include, but are not limited to, smart computers, cellphones, office supplies, wirelessly enabled cars, lighting systems, HVAC (heating, ventilation, and air conditioning), and household appliances. A device (or "thing") must be on a network and connected to a communication node in order to be IoT-enabled. Connectivity services for IoT deployment on several platforms are offered by a variety of communication network technologies (infrastructures), including 3G, LTE, Wi-Fi, Bluetooth, ZigBee, Z-wave, Sigfox, and others.

Statement of the Problem

The rapid proliferation of IoT-enabled Android devices has significantly increased the attack surface for ransomware. Existing network- and signature-based intrusion detection systems often fail to account for the unique, time-series energy consumption patterns of malicious code, resulting in security solutions that are either too resource-intensive for constrained devices or insufficiently accurate in early detection.

Purpose of the Study

The purpose of this study is to develop and evaluate a machine-learning framework that leverages Android device power consumption logs to accurately distinguish ransomware from benign applications, thereby enabling lightweight yet effective intrusion detection on resource-constrained IoT nodes.

Research Questions

The following research questions guided the study;

1. **Energy-Based Differentiation**: Can distinct power-usage fingerprints reliably differentiate ransomware from benign apps on Android devices?

2. **Classifier Performance**: Which supervised learning algorithm (KNN, Neural Network, SVM, and Random Forest) achieves the highest accuracy, precision, recall, and F-measure when applied to normalized power-usage subsamples?

3. **Optimal Sampling Window**: What sampling window size maximizes detection performance metrics for time-series classification of energy consumption?

Materials and Methods

In order to identify and categorize attacks in IEEE 802.11 networks, Thing (2019) examined the dangers to these networks and suggested an anomalous network intrusion detection system. For the IEEE802.11 standard, this study is regarded as the first to use deep learning methods. Thing experimented with two and three hidden layers in the Stacked Auto-encoder (SAE) architecture. The author experienced distinct activation functions for the buried neurons. He assessed his method using a dataset generated from a lab-emulated Small Office Home Office (SOHO) architecture. He achieved an overall accuracy of 98.66% in a 4-class classification (i.e., legitimate traffic, flooding type assaults, injection type attacks, and impersonation attacks). In order to identify intrusions, Diro et al. (2022) suggested adopting fog computing in IoT systems.

By supplying the fog layer (hubs, routers, or gateways) with sophisticated intermediate data processing, fog computing aims to boost productivity and reduce the volume of data transmitted to the cloud. Such technology, which is more efficient in terms of scalability, autonomy in local attack detection, acceleration on data training near sources, and sharing of nearby parameters, enables distributed attack detection. The authors proposed a deep learning technique to detect both known and unknown intrusion risks. Given that 99 percent of attacks are known, it may be said that zero-day attacks are made by making minor changes to known ones. Therefore, multi-layer deep networks enhance minor change awareness (in a self-taught manner with compression capabilities) compared to shallow learning classifiers.

The basis of the distributed deep learning approach is the rapid and local distribution of the dataset to train each sub-dataset, followed by the sharing and coordination of learning parameters with neighbors. As a result, the design ends with a master IDS that modifies the parameters of the down-dispersed IDSs while maintaining synchronization. The studies show that the accuracy of the distributed parallel deep learning technique is higher than that of the centralized deep learning NIDS and shallow machine learning techniques. To train the models and evaluate the IDS, Diro et al. altered the NSL-KDD dataset to contain 123 input features and 1 label.

As a result, they achieved 96.5% detection rate and 2.57% false positive rate using multi-class detection with four labels (normal, DoS, Probe, and R2L.U2R). A novel fog computing-based intrusion detection method with a distributed security mechanism that respects the interoperability, flexibility, scalability, and heterogeneity aspects of

IoT systems was proposed by Prabavathy et al. (2021) using Online Sequential Extreme Learning Machine (OS-ELM). The alerts for the deep model were 93.66% detection and 4.97% false detection rate, and they also found an increase in the total detection accuracy as the number of fog nodes grew from 96% to 99%. The two primary parts of the suggested system are as follows:

1. **Attack detection at fog nodes**: Prabavathy et al. use OSELM algorithm to detect intrusions in fog nodes. The IoT network is divided into virtual clusters where each cluster corresponds to a group of IoT devices under a single fog node. The OSELM classifies the incoming packets as normal or an attack. ELM is a single hidden layer feed forward neural network characterized by its fast-learning phase. The input layer weights and hidden layer bias values are randomly selected to analytically deduce the output weights using simple matrix computations. However, the online nature of OS-ELM favors a streaming detection of IoT attacks.

2. **Summarization at cloud server**: to have a general idea about the global security state of the IoT system, detected intrusions are sent from the fog node to the cloud server. After the analysis and the visualization of the current state, Prabavathy et al. propose two actions;
   i. predict next attacker action using the attacker plan recognition approach; or
   ii. identify fog node geographical position based multistage, and DDoS attacks.

Hence, an intrusion response can be activated.

To test their hypothesis, he also recommended a proof of concept that used a DUALCORE processor, 200 GB HDD, and 1 GB RAM as fog nodes. The authors utilized an Azure cloud service with four AMD Opteron 2218 dual-core processors (2.6 GHz, 8 core, 32 GB RAM, and 6146 GB HDD) for the experimental setup. They used NSL-KDD as a benchmark dataset and MATLAB to create OS-ELM. The authors claimed high accuracy and response time. They have a 97.36% accuracy rate and a 0.37% lower false alarm rate. The fog node technique had a 25% higher detection rate than cloud-based deployment. One important advantage is that new online data can be incorporated into the learning process, unlike ANN and NB.

Methods

We must first record the power usage of the targeted applications in order to create a fingerprint of the ransomware's energy consumption. In line with earlier research methods (Yang 2012; Merlo et al. 2015), we employed Power-Tutor to track and sample the power consumption of every process that was operating at 500 ms intervals. PowerTutor generates log files that show the energy consumption of every process at a specified sample interval. A Samsung Galaxy SIII (CPU: 1.4 GHz, RAM: 2GB, OS: Android 4.4), a Samsung Galaxy S Duos (CPU: 1.0 GHz, RAM: 768MB, OS: Android 4.0.1), and an Asus Padfone Infinity (CPU: 1.7 GHz, RAM: 2 GB, OS: Android 4.4) were the three Android devices on which we ran our tests. To collect energy consumption logs of both ransomware and goodware, we installed the most popular Android applications, namely:

Gmail (version 9.6.83), Facebook (version 99.0.0.26.69), Google Chrome (version 53.0.2785.124),

Youtube (version 11.39.56), Whatsapp (version 2.16.306), Skype (version 7.20.0.411), AngryBrids (version 6.1.5), Google Maps (version 9.39.2), Music Player (version 4.2.52), Twitter (version 6.19.0),

six latest and live ransomware strains on all platforms, as well as Instagram (version 9.6.0) and Guardian (version 3.13.107). All malware had active Command and Control (C2) servers and were downloaded using the VirusTotal 1 Intelligence API. After then, we utilize PowerTutor to track and log the power consumption of the device processes for five minutes while the malware and apps are operating independently. The user interactions when using the programs (also known as goodware) were similar to those in the real world. Each gadget underwent this process five times, therefore we were able to

5repeation × 3device = 15 power usage samples for each and every application and ransomware.

In order to have a valid evaluation, the energy consumption of all devices was mapped to a defined range because each device's CPU has its own power usage specification. Since 0 denotes no power usage and 1 denotes the maximum CPU power utilization, we normalized the CPU power consumption for all observed processes on the devices to [0, 1]. Log files were processed, power usage numbers were extracted and standardized, and a row-normalized dataset was produced using scripts. A label (such as ransomware or goodware) and a normalized sequence of energy usage for five minutes of activity are included in each row.

Classification of the Algorithm Used

A crucial component of Supervised Learning and Classification is giving a sample the appropriate label based on prior observations (Michalski et al. 2013). To identify the class of each power consumption sequence, we used four cutting-edge classifiers on the power usage samples: kNearest Neighbor (KNN), Neural Network (NN), Support Vector Machine (SVM), and Random Forest (RF). KNN is a straightforward and effective classifier that finds the K nearest sample or samples and labels the provided samples with the majority of their neighbors. Human brain networks are implemented by NN (Haykin 1998), which is primarily used to approximate the function between inputs and outputs.

SVM, another well-liked supervised learning method, is predicated on the idea of decision planes that provide decision bounds. A decision plane distinguishes between a group of items according to the classes they belong to. RF (Verikas et al. 2011) was developed with ensemble learning as its driving force. It generates the class label by building a large number of decision trees during training.
Each process's power usage sequence can be thought of as time-series data. To categorize time-series data, numerous approaches have been put forth (Xing et al. 2010). In this study, a distance-based time-series classification approach based on Dynamic Time Warping (DTW) (Müller et al., 2023) is used for distance measure, and KNN is used as a classifier. Similarity distance is a key element in KNN classification and we apply two different distances to find the closest neighbor as follows:

i. Euclidean distance: Euclidean distance or Euclidean metric is the intuitive distance between two vectors in Euclidean space and calculated as follow:
ii. Dynamic time warping (DTW): DTW is a recognized technique for finding an optimal alignment between two time-dependent sequences (see Fig. 1). According to DTW's ability to deal with time deformations and issues associated with speed differences in time-dependent data, it is also employed to calculate distance or similarity between time series (Müller et al., 2022). Let us denote two sequences that display two discrete subsamples as

X = (x1, ... , xn) and Y = (y1, ... , ym) of length m, n ϵϵ ℕ.

A Cost Matrix Cϵϵℝn×m is used by DTW. The distance between xi and yj is shown in each cell Ci,j (see Fig. 2). The goal of DTW is to find the best alignment between X and Y with the least amount of total distance. To put it simply, an ideal alignment passes through a valley of inexpensive cells in cost matrix C. A sequence p = {p1,..., pL} with pl = (nl, ml)ϵϵ[1:N] × [1:M], lϵϵ[1:L] that satisfies the following criteria is used to specify a warping path:

- Boundary condition: p1 = (1, 1) and pL = (N, M).
- Monotonicity condition: n1 ≤ n2 ≤ ⋯ ≤ nL and m1 ≤ m2 ≤ ⋯ ≤ mL.
- Step size condition: pl+1 − pl = {(1, 0), (0, 1), (1, 1)} for l[1:L1].

The summing of all local distances of a warping path's elements outcomes the overall cost of path and in order to discover optimal warping path p∗, the path having minimal total cost among all feasible pathways is selected. Lastly, the total cost of the best warping path is calculated for two sequences, X and Y, to determine how similar or different they are. In relation to the local cost measure c, the total cost cp(X, Y) of a warping path p between X and Y is defined as follows:

(2)cp(X, Y) =Ll=1 c(xnl, yml).

The DTW distance DTW (X, Y) between X and Y is then defined as the total cost of p∗:

Figure 3 illustrates how DTW aligns two power usage subsamples in order to find optimal path between them for distance calculation.

Metrics and Cross-Validation

We employ the following four widely used performance metrics for malware identification, which is comparable to the methodology in Buczak and Guven (2021): When ransomware is accurately identified as a malicious application, it is referred to as a true positive (TP). When a goodware program is accurately identified as a non-malicious application, it is said to be true negative (TN). A false positive (FP) occurs when a malicious application is incorrectly identified as goodware.

A ransomware detection that is labeled as a non-malicious application is known as a false negative (FN).

To evaluate the effectiveness of our proposed method, we used machine learning performance evaluation metrics that are commonly used in the literature, namely: Accuracy, Recall, Precision and F-Measure.

Accuracy is the number of samples that a classifier correctly detects, divided by the number of all ransomware and goodware applications:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN.}$$

Precision is the ratio of predicted ransomware that are correctly labelled a malware. Thus, Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall or detection rate is the ratio of ransomware samples that are correctly predicted, and is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN.}$$

F-Measure is the harmonic mean of precision and recall, and is defined as follows:

$$F - \text{Measure} = \frac{2 \ast TP}{2 \ast TP + FP + FN}$$

One essential machine learning technique for determining how well experiment results may be extrapolated to a separate dataset is cross-validation (Kohavi et al., 1995). We employed leave-one-out cross validation to assess the effectiveness of the suggested approach. We know that all subsamples of a sample must be removed from the classifier training phase in order to apply this validation technique. A Microsoft Windows 10 Pro PC with an Intel Core i7 2.67 GHz processor and 8 GB of RAM was used for all experiments, running MATLAB R2015a.

Results

This section presents a summary of the dataset used for model training and evaluation. Table 1 details the hardware specifications of the Android devices.

Table 1. Dataset Summary

Device	CPU (GHz)	RAM	OS Version
Samsung Galaxy S III	1.4	2 GB	Android 4.4
Samsung Galaxy S Duos	1.0	768 MB	Android 4.0.1
Asus Padfone Infinity	1.7	2 GB	Android 4.4

However, as patterns of power consumptions are not predictable and depend on many factors such as files content, encryption algorithm etc. samples are highly distributed in the feature space. It appears that direct application of conventional classification algorithms namely NN, KNN and SVM, is not promising. For example, the KNN classifier that uses DTW as a similarity measure outperformed other techniques while conventional KNN (with parameter setting of K = 1, 5, 10) is ranked lowest among the classification approaches.

Since Euclidean method calculates similarity by summing distances between corresponding points of samples, the calculated distance could be far when the position of occurring power usage patterns varies (even if samples are visually cognate). On the other hand, DTW attempts to align samples based on the distance between pieces of samples that are more similar regardless of the position of similar energy usage pattern. Consequently, the performance of KNN classifier is significantly influenced by the distance criteria. The second place belongs to RF that selects subset of features and works in splitted feature spaces instead of using a complete feature space. These observations led us to hypothesis that a subset of features (i.e., a specific interval within Ransomware infection period) may improve performance of the classification techniques.

Table 2: performance Evaluation of the Ransomware using different algorithms

Algorithms	**Accuracy (%)**	**Recall (%)**	**Precision (%)**	**F-measure (%)**
KNN (k = 1)	71.85	71.11	56.14	62.75
KNN (k = 5)	72.59	72.22	57.02	63.73
KNN (k = 10)	83.79	71.11	56.64	63.05
KNN (k = 1 and DTW)	83.79	78.89	73.96	76.34
Neural network	75.93	73.33	61.68	67.01
Random forest	80.74	76.67	69.00	72.63
SVM	78.52	74.44	65.69	69.79

Table 3: Evaluation metrics for different window sizes and SVM: a comparative summary

Data	Accuracy (%)	Recall (%)	Precision (%)	F-measure (%)
5	77.72	59.42	73.21	65.60
10	88.60	85.51	83.10	84.29
15	91.19	**94.20**	83.33	**88.44**
20	**89.64**	82.61	**87.69**	**85.07**
25	87.56	75.36	88.14	81.25
30	81.35	55.07	**88.37**	67.86
35	78.24	47.83	84.62	61.11
40	78.24	47.83	84.62	61.11
45	76.17	42.03	82.86	55.77
50	76.68	42.03	85.29	56.31

Best (optimal) values are highlighted in bold

Table 4: Evaluation metrics for different window sizes and neural network: a comparative summary

Data	Accuracy (%)	Recall (%)	Precision (%)	F-measure (%)
5	88.08	82.61	83.82	83.21
10	88.08	84.06	82.86	83.45
15	**89.64**	**88.41**	83.56	**85.92**
20	90.67	86.96	86.96	86.96
25	89.64	85.51	85.51	85.51
30	89.12	85.51	84.29	84.89
35	88.08	82.61	83.82	83.21
40	86.01	81.16	80.00	80.58
45	85.49	82.61	78.08	80.28
50	86.01	82.61	79.17	80.85

Best (optimal) values are highlighted in bold

As shown in Table 3, the KNN classifier that uses DTW distance with a subsample size of 7.5 s outperformed all other methods in terms of detection rate95.65% and performance of 94.27%. Although KNN is the least sophisticated classification approach, it outperformed other rival classification techniques since it only relies on the formation and distribution of goodware's and ransomware's subsamples.

The performance of KNN using DTW for all evaluation metrics peaks at window size = 15. However, the remaining classifiers were not able to achieve an optimal performance at the specified window size. For example, NN's best accuracy, precision and Fmeasure occurred at w = 20, while highest recall was achieved at w = 15. The numerical results indicate that subsamples are not from specified and exact data distribution and classes have overlap sample(s) in feature space. Therefore, KNN that seeks for most similar subsample to input data outperform other classification approaches.

Moreover, according to ability to align subsamples, DTW can find closer energy consumption pattern and consequently provide more accurate classification results than euclidean. Furthermore, and in practice, KNN's requirement for concurrent distance calculations between training and testing objects can be implemented using parallel processing (so distances can be independently computed). Subsamples dictionary can be partitioned into sperate IoT nodes and each subsample is sent to nodes. They return a label and a similarity value and the label having less similarity value is final subsample's label. This approach reduces the classification time and mitigates the need for storage capacity in every node.

Conclusion

With increasing proliferation of Internet linked devices and things in our datacentric culture, protecting the security of IoT networks is crucial. Successfully compromised IoT nodes could hold the network to ransom significantly impair the operation of a company and result in considerable financial loss and reputation harm.

In this work, we proposed a method for identifying ransomware based on its power usage. In particular, we differentiate ransomware from benign apps by using the distinct local fingerprint of ransomware's energy usage. The sequence of applications' energy consumption is splitted into numerous sequences of power usage subsamples, which are then classed to produce aggregated subsample's class labels. Our trials showed that our method produced a precision rate of 89.19 percent and a detection rate of 95.65%.

Future works include prototyping the proposed approach for deploying in a real world IoT network, with the aims of evaluation and refinement.

Recommendations

1. Real-World Prototyping: Deploy the proposed energy-consumption-based detection model in a live IoT network to validate operational feasibility and robustness under real traffic conditions.
2. Parameter Refinement: Adopt a 15-second subsample window with DTW-based KNN—identified as optimal in our experiments—to balance detection accuracy (≈91%) and responsiveness.
3. Distributed Processing: Leverage parallel distance calculations across fog or edge nodes to reduce classification latency and minimize per-node storage requirements.

References

Ali, B., & Awad, A. I. (2021). Cyber and physical security vulnerability assessment for IoT-based smart homes. *sensors*, *18*(3), 817.

Caceres, R., & Friday, A. (2021). Ubicomp systems at 20: Progress, opportunities, and challenges. *IEEE Pervasive Computing*, *11*(1), 14-21.

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.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2022). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645-1660.

Karimi, K., & Atkinson, G. (2023). What the Internet of Things (IoT) needs to become a reality. *White Paper, FreeScale and ARM*, 1-16.

Lai, C. F., Lai, Y. X., Yang, L. T., & Chao, H. C. (2019, November). Integration of IoT energy management system with appliance and activity recognition. In *2012 IEEE* International *Conference on Green Computing and Communications* (pp. 66-71). IEEE.

Li, Z., Liu, G., Liu, L., Lai, X., & Xu, G. (2018). IoT-based tracking and tracing platform for prepackaged food supply chain. *Industrial Management & Data Systems*.

Merlo, A., Migliardi, M., & Caviglione, L. (2019). A survey on energy-aware security mechanisms. *Pervasive and Mobile Computing*, *24*, 77-90.

Müller, J. M., Erdel, M., & Voigt, K. I. (2018). Industry 4.0-Perspectives and challenges for project logistics. In *EurOMA Conference, Edinburgh, Scotland*.

Prabavathy, S., Sundarakantham, K., & Shalinie, S. M. (2020). Design of cognitive fog computing for intrusion detection in Internet of Things. *Journal of Communications and Networks*, *20*(3), 291-298.

Roman, R., Zhou, J., & Lopez, J. (2021). On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, *57*(10), 2266-2279.

Xing, L., Tannous, M., Vokkarane, V. M., Wang, H., & Guo, J. (2023). Reliability modeling of mesh storage area networks for Internet of Things. *IEEE Internet of Things Journal*, *4*(6), 2047-2057.