sustainability

MDPI

Article

A Trust-Based Secure Parking Allocation for IoT-Enabled Sustainable Smart Cities

Javed Ali [1] and Mohammad Faisal Khan [2,*]

[1] College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia; j.ali@seu.edu.sa
[2] Department of Basic Sciences, College of Science and Theoretical Studies, Saudi Electronic University, Riyadh 11673, Saudi Arabia
* Correspondence: f.khan@seu.edu.sa

Abstract: Smart parking is a crucial component of smart cities that aims to enhance the efficiency and sustainability of urban environments. It employs technology such as sensors and IoT devices to optimize the use of parking resources and improve drivers' experiences. By reducing traffic congestion, decreasing air pollution, and enhancing accessibility, smart parking systems can contribute to the overall well-being of urban areas. IoT-enabled smart parking refers to the application of IoT technology to optimize and improve parking efficiency in smart cities. However, security and privacy challenges in IoT-enabled smart parking pose risks and concerns related to the collection and use of data by parking systems, such as unauthorized access or misuse of data, potential data breaches, and the need to ensure responsible data collection and usage to maintain user trust and confidence. To address these challenges, we propose a novel hybrid approach to trust management using machine learning algorithms to enhance the security and privacy of the system. Our approach consists of SVM and ANNs, taking into account credibility, availability, and honesty as key parameters. Furthermore, we use ensemble machine learning to select the best-predicted model from different trained models, leading to efficient performance and a trustworthy environment. Our results show that the proposed hybrid SVM classifier with a trust parameters approach achieved an accuracy of 96.43% in predicting and eliminating malicious or compromised nodes.

Keywords: Internet of things; smart parking; sustainable smart cities; security; trust management; privacy preservation; trustworthiness

Citation: Ali, J.; Khan, M.F. A Trust-Based Secure Parking Allocation for IoT-Enabled Sustainable Smart Cities. *Sustainability* **2023**, *15*, 6916. https://doi.org/10.3390/su15086916

Academic Editors: Mohammed Elsayed Lotfy, Iqram Hussain, MD Rashedul Hasan Sarker and Md Azam Hossain

Received: 4 March 2023
Revised: 8 April 2023
Accepted: 13 April 2023
Published: 20 April 2023

1. Introduction

Smart cities are urban areas that use technology to improve the efficiency and sustainability of their assistance and infrastructure. By incorporating information and communication technologies (ICT) [1] and the Internet of things (IoT) [2], smart cities optimize resource utilization, decrease waste and pollution, and enhance the quality of life for their residents. Sustainable smart cities [3], a subset of smart cities, specifically concentrate on advancing environmental sustainability and decreasing the negative effect of urbanization on the atmosphere. This includes efforts to reduce greenhouse gas emissions, minimize waste and pollution, and protect natural resources. To accomplish these goals, sustainable smart cities often depend on various technologies and strategies, such as energy-efficient buildings [4], renewable energy sources [5], smart transportation systems [6], and data analytics to optimize resource usage [3]. IoT-enabled [7] sustainable smart cities are urban areas that use IoT technology to improve the efficiency, sustainability, and livability of their benefits and infrastructure [8]. By integrating IoT devices and sensors into the constructed surroundings, these metropolises can optimize resource usage, diminish waste and pollution [9], and enhance the quality of life for their nationals. Examples of IoT-enabled sustainable smart cities include smart meters [10] to optimize energy usage in buildings,

sensors to observe air quality and traffic patterns, and IoT-enabled waste management systems [11] to lessen waste and improve recycling speed.

Despite the considerable advantages of sustainable smart cities, there are also substantial challenges that must be overcome to implement these initiatives successfully. One of the main challenges is the need for extensive planning and coordination between different stakeholders, including city administrations, businesses, and locals. Data privacy and security [12] are also critical concerns in sustainable smart cities, as more additional data are gathered and transmitted through IoT devices and sensors. Securing that collected data, and storing it responsibly is fundamental for building trust in these initiatives. The car parking system [13] incorporates IoT devices, which are known for their heterogeneous structure. One challenge associated with IoT is the potential for malicious nodes to transform sensor data, leading to mismanagement of the process at the base station and disabling the entire sustainability of the parking system. To address this issue, additional computation energy is required to detect and eliminate malicious nodes. In order to secure the system, a novel methodology has been developed to improve the performance of the smart parking system. This article proposes a novel trust-based approach for identifying malicious and compromised nodes in order to maintain a trustworthy environment in smart cities. Another goal of the proposed approach is to secure the process of allocating parking slots to valid nodes in a way that maximizes the efficient utilization of available slots. The proposed approach employs an RSU and a parking recommender to calculate the trustworthiness of nodes and maintain a table for periodic slot allocation. The proposed trust management system aims to enhance the security and efficiency of the parking allocation process in smart cities. The novel contributions of the proposed mechanism can be summarized as:

1. A trust-based approach is developed to provide the security of the parking allocation procedure and to protect the privacy of vehicles and their car locations.
2. The use of novel parameters authorizes the effective identification of malicious nodes, ensuring that the allocation of parking slots is not disrupted.
3. The maintenance of trust levels of nodes allows for the prioritization of parking slot allocation to ensure that slots are distributed fairly and efficiently.
4. The dissemination of a priority table using roadside units to enables nodes to efficiently access information about available parking slots and prioritize their requests based on their trust levels.

The structure of the proposed paper is as follows. In Section 2, we discuss the relevant research on trust management in smart parking systems. Section 3 covers the approaches for implementing a trust-based secure parking allocation system for IoT-enabled sustainable smart cities, including the various trust-based methodologies evaluated for securing car parking. Section 4 presents a performance comparison of the proposed approach with existing mechanisms. Finally, in Section 5, we provide a conclusion for the paper.

2. Related Work

IoT-enabled smart parking systems have led to the development of various approaches for optimizing the allocation of parking slots and improving the efficiency of parking in urban environments. This section discusses and compares the various approaches that have been proposed in the literature for smart parking slot allocation, with a focus on those that address the challenges of maintaining data privacy and security [14]. The section also provides a comparative analysis of these approaches (as shown in Table 1), highlighting their strengths and limitations, and discusses how they contribute to the overall goal of improving the efficiency and sustainability of smart parking in smart cities.

Table 1. Comparative Analysis of the Literature Review.

Ref.	Main Focus	Limitation
[15]	IoT-Based E-Parking System	No mention of malicious nodes
[16]	Parking System Using RFID	No real-time data processing
[17]	IoT Assisted Intelligent Parking System	Limited coverage area
[18]	Fog-Blockchain Computing for Autonomous-Vehicle Parking	Complexity of implementation
[19]	Energy Management System of Campus Microgrids	Not solely focused on parking systems
[20]	Privacy-Preserving Smart Parking System	Limited scalability
[21]	Smart Parking Using IoT Technology	No mention of malicious nodes
[22]	Smart Parking System with Privacy Preservation	Complexity of implementation
[23]	Intelligent Approach for Smart Car Parking	No real-time data processing
[24]	IoT-Enabled Trust-Based Secure Wireless Energy Sharing	Not solely focused on parking systems
[25]	Intelligent Parking Sharing System for Green and Smart Cities	Limited scalability
[26]	Trustworthy Parking Communities	Limited scalability
[27]	SmartParking: A Secure and Intelligent Parking System	No mention of malicious nodes
[18]	Real Time Car Parking System	No mention of malicious nodes

An IoT-based e-parking system for multiplexes and shopping malls was suggested in 2023. The system uses sensors to detect free parking spaces and uses a mobile application for booking and payment [15]. Similarly, Sheng et al. developed a parking system using RFID (radio-frequency identification) technology [16]. The system allows drivers to find available parking spaces and facilitates payment by RFID tag. On the other hand, Aditya et al. proposed an intelligent parking system "IPS" [17]. A smart city using Internet of things (IoT) technology. The system uses sensors to detect parking space availability and provides real-time information to drivers via a mobile application. Additionally, Shahzad et al. proposed an autonomous parking system [18] based on fog blockchain computing. The system utilizes a blockchain-based secure data storage approach and fog computing to enable efficient and secure parking for autonomous vehicles. In [19], the authors present a review of the state-of-the-art issues and potential problems with creating and implementing campus microgrid energy management systems. The main elements and purposes of energy management systems are covered by the authors, along with various control methods and optimization strategies that can be applied to boost the effectiveness and dependability of microgrid systems. The article also discusses some of the main opportunities and difficulties involved in integrating smart grid, energy storage, and renewable energy technologies into campus microgrids, as well as some potential solutions to these difficulties.

An approach was proposed in 2016 to maintain privacy in smart parking using elliptic curve cryptography to create a secure platform [20]. The study noted that security and privacy issues were raised due to the limited capabilities of devices in wireless communication. The proposed approach used elliptic curve cryptography as an alternative to traditional encryption approaches to increase efficiency in terms of resource utilization, such as reduced processing power and memory requirements, and lower energy consumption. The proposed approach aimed to address four major challenges: platform independence, exchangeability, OS independence, scalability, and efficiency. The contribution of the proposed approach was to secure the transmission and provide privacy and data integrity using a lightweight cryptography approach along with the use of a zero-knowledge protocol on discrete algorithms. However, the proposed approach can generate a limited number of keys in the library, which are needed to fulfill the requirements.

In 2018, an article was proposed that aimed to make advancements in smart parking systems by using a first-time computer vision technique to identify the plate number and parking slot [21]. Digital payment was also introduced in the smart parking system, and the security of the system was improved. The proposed approach also included a feature to help users locate their car if they forget where they parked it. The architecture of the proposed approach involved the use of smart parking sensors connected to a Raspberry PI, with data being sent to cloud computing software to facilitate the user. However, the proposed approach has limitations, including the need for further improvements in smart

parking techniques and the potential for cloud data to be altered by hackers. The article "Mez" proposes an adaptive messaging system, built on top of the MQTT protocol, for multi-camera machine vision applications at the edge of the IoT network. The system is designed to address the challenges of high data volume, high data rate, and real-time processing requirements [28]. The Mez messaging system consists of the Mez server, the Mez broker, and the Mez client, and is adaptive and dynamic, allowing it to adjust message size and frequency based on network conditions and processing capabilities. The article highlights the importance of low-latency messaging systems for machine vision applications in the IoT edge, and the Mez system's ability to improve performance and enable real-time decision making is a significant contribution to the field.

In 2020, a new approach for improving smart parking systems through the use of computer vision and digital payment methods [22]. The system, built on a network of smart parking sensors connected to a Raspberry PI and accessed through cloud-based software, aims to increase security and convenience for users. While the system has demonstrated significant advancements, it is not yet fully secure, as there is potential for data tampering by hackers. Despite this limitation, the proposed approach shows promise for further development and improvement in smart parking technology. In 2014, the implementation of blockchain technology was used to improve the security and transparency of a smart parking system [23]. This decentralized approach, which incorporated anonymous credentials and a focus on preserving user privacy, helped to secure the system against potential attacks [29] and increase public trust. One challenge in using blockchain for parking systems is the potential for private parking owners to share their slots when not in use, which could increase the number of available spaces but also raise concerns about privacy. It is important to address these issues in order to maintain the integrity and effectiveness of the system.

In 2021, a paper proposed a solution to improve the performance of intelligent parking systems using IoT and game theory [24]. The proposed model was compared to the ASPIR model and demonstrated improved performance. The architecture of the proposed model incorporates game theory in order to address challenges in finding the best parking location and determining available spaces. However, the model does have limitations, including difficulties in knowing the distance of available parking spaces from a destination and the lack of information about the walking distance from a parking stand. In 2015, an article proposed the use of "Parking Communities" to facilitate the process of finding parking spaces for vehicles [25]. This approach is based on encryption and signature algorithms, as well as a mathematical trust rating model, and was found to have higher performance compared to other security architectures. The architecture of the system includes creating community trust, querying, responding, and rating. One challenge associated with this approach is the need to improve resource management and prioritize incoming queries based on various factors, such as energy or response budgets or verifiable properties such as disability certificates.

In 2008, a secure and intelligent parking system was proposed using the framework of NOTICE, a secure and privacy-aware architecture for traffic incident notifications [26]. The system, called SmartParking (SP), allows drivers to view and reserve parking spaces in a service-oriented manner, streamlining the parking process. The proposed system is designed to protect against security and privacy attacks, and the authors discuss hardware and software implementations as well as the overall architecture. One limitation of the system is that it may take longer to search for a parking space when the number of vehicles is high. In 2017, a study was conducted to examine the integration of various security measures for the protection of vehicles in parking lots, aiming to make the system more reliable and efficient [27]. The use of the IoT as part of the edge layer and fog nodes offers several benefits and flexibility in parking. The proposed architecture includes a network fog center for the SCPS model, as well as a microprocessor and various sensors. Future studies will focus on analyzing the issue of live migration of virtual machines (VMs) to

increase energy efficiency at fog nodes in a vehicular cloud data center (VCD) that handles large datasets of traffic.

In 2022, a lightweight integrated blockchain and cryptography module was developed to authorize and grant access to autonomous vehicles (AVs) in each segment of the parking system [18]. The module operates at each fog node and is composed of a fog blockchain computing perception layer, an intermediate fog layer, and a cloud layer. The proposed approach aims to improve upon current smart parking (SP) systems and presents a comprehensive, long-term, effective, and well-performing smart autonomous vehicle parking (SAVP) system that utilizes emerging fog-computing and blockchain technologies as robust solutions to enhance the collaborative IoT-cloud platform for building and managing SP systems for AVs. One limitation of the proposed approach is the need for continued development of SP systems.

3. Proposed Methodology

Smart cities are an important focus of research in the modern world, as we seek to create sustainable and efficient urban environments. One of the key challenges in smart cities is parking allocation, as the increasing number of vehicles on the roads has led to a shortage of parking spaces in many urban areas. To address this challenge, we propose a trust-based secure parking allocation mechanism for IoT-enabled smart cities. The proposed architecture for our approach consists of three major components: the base station, smart parking, and roadside unit (RSU), as illustrated in Figure 1. The vehicle can communicate with the RSU to request parking allocation, and the RSU computes the trust of a node using trust parameters such as credibility, availability, and honesty. Nodes with a higher degree of trust will be given priority for parking allocation, and all computations performed will be stored on the base station for future use.

The trust degree is calculated as a single value ranging from 0.0 to 1.0. After evaluating all three trust parameters, the aggregated trust value is compared to a default threshold of 0.5. If the value is greater than 0.5, the node is considered trustworthy, and if the value is less than 0.5, the node is deemed non-genuine, which implies that it is a malicious node that may disrupt the system. Our approach also proposes a novel method that integrates a hybrid ensemble learning approach using an SVM classifier and a trust management approach. This approach helps to establish trust between vehicles, parking stands, RSUs, and the base station. The proposed methodology is centralized, which means that all trust calculations and parking allocation decisions are made at the base station. This helps to ensure that the system is secure and that there is no risk of malicious nodes disrupting the system.

The base station is the central component of the architecture and is responsible for storing all the data related to parking allocation and trust management. It is also responsible for processing all the requests for parking allocation from the vehicles and for communicating with the RSUs to allocate parking spaces. The smart parking component is responsible for managing the parking stands and for communicating with the RSUs to provide information about the availability of parking spaces. The RSUs are responsible for communicating with the vehicles and for computing the trust of a node based on the trust parameters. They are also responsible for communicating with the smart parking component to request parking allocation for the vehicles. The trust parameters used in our approach are credibility, availability, and honesty. Credibility refers to the degree to which a node can be trusted to provide accurate information. Availability refers to the degree to which a node is available to participate in the system. Honesty refers to the degree to which a node is trustworthy and engages in honest behavior without engaging in malicious activity. Our approach also includes a machine learning component that uses an SVM classifier to predict the trustworthiness of a node based on historical data. This approach helps to improve the accuracy of the trust calculations and helps to identify malicious nodes more quickly.

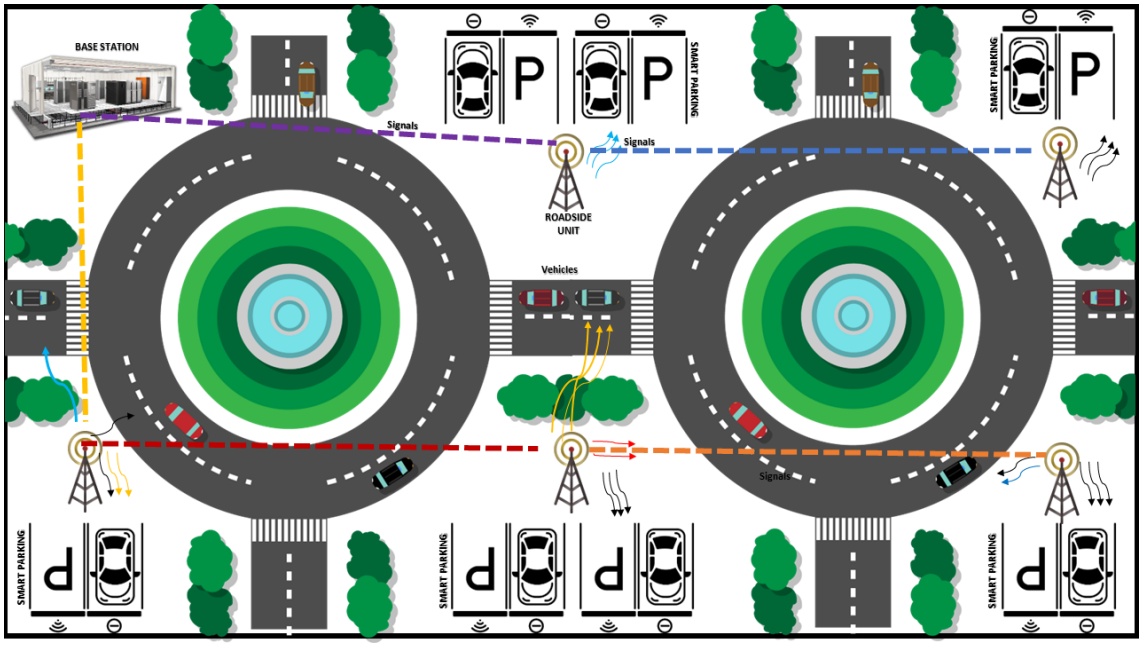

Figure 1. The Proposed Architecture For a Smart Parking System.

3.1. Working of Proposed Methodology

Vehicles park, as shown in Figure 1, and will send requests to the RSU, which retrieves the status of nearby parking stands. The RSU also sends all information to the base station. All the RSUs send information to the base station, which is responsible for making final decisions. If a parking stand is full, the RSU sends a request to the base station, which checks the status of nearby RSUs and sends the location of another nearby parking stand to the RSU. The RSU then sends the information directly to the vehicle, which parks at the new location. If a malicious node tampers with any RSU data, it could result in vehicles parking in the wrong locations. This not only wastes fuel but also contributes to rising fuel costs. To mitigate this issue, we are proposing a trust management methodology using ensemble machine learning. The aim of this article is to eliminate malicious nodes in the parking system. To achieve this, the proposed approach first establishes trust between the vehicles and RSU, between the parking stands and RSU, and between the RSU and the base station. The complete working flow of trust computations and decision making is represented in Algorithm 1.

The proposed Algorithm 1 is aimed at computing the trust degree of each node in the network based on the observations' weight of credibility, honesty, and availability. The trust degree calculated for each node is a weighted sum of these three parameters, where the weights are set based on the observation made in the network. The algorithm starts by initializing an empty dictionary to store the trust degree of each node. This trust degree dictionary is defined as $trust\_degree = d_f$, where d_f is the trust degree of node f in the network.

Next, the algorithm proceeds to compute the trust degree of each node in the network based on the three parameters, credibility, honesty, and availability. For each node in the network, the algorithm calculates these three parameters using the functions $compute\_credibility$, $compute\_honesty$, and $compute\_availability$, respectively. Once the credibility, honesty, and availability parameters have been computed for each node, the trust degree is then calculated as a weighted sum of these parameters. The weight of each parameter is defined as the observation weight of that parameter, which is set based on the observations made in the network.

Algorithm 1 The Computational Process Computing the Trust Degree of a Node

Input: Nodes present in the network.
Input: Observations weight of credibility.
Input: Observations weights of honesty.
Input: Observations weight of availability
Output: Trust degree sorting.

procedure TRUST EVALUATION
    Initialize an empty dictionary to store the trust degree of each node
    $trust\_degree = \{d_f\}$
    For each node in the network, compute the trust degree based on the credibility, honesty, and availability parameters. The trust degree is calculated as a weighted sum of these three parameters.
    `for node in the network:`
    credibility = compute_credibility(node)
    honesty = compute_honesty(node)
    availability = compute_availability(node)
    trust_degree = credibility_weight * credibility + honesty_weight * honesty + availability_weight * availability
    Sort the trust degrees in descending order and create a priority table based on the trust degrees.
    sorted_trust_degrees = sorted(trust_degrees.items(), key=lambda x: x[1], reverse=True)
    Return to the priority table.
Exit.

After the trust degree has been computed for each node in the network, the trust degrees are sorted in descending order, and a priority table is created based on the trust degrees. This priority table is used to allocate parking spaces to nodes in the network based on their trust degree, where nodes with higher trust degrees are given priority. Finally, the algorithm exits after returning the priority table, which can be used for parking allocation in the smart city environment. The notable aspect of Algorithm 1 is that it is designed to be run periodically to continuously update the trust degree of nodes in the network based on their behavior and observations made in the network. This ensures that the trust degree of nodes is updated regularly and reflects their current behavior in the network. The computation of trust and decision making is demonstrated in Algorithm 2.

Algorithm 2 Decision Making of Malicious and Trustworthy Nodes Based on Trust Degree

Input: - a node in the network
Output: - trust_degree: the trust degree of the node (trust value between 0.0 and 1.0)

Procedure
credibility = Compute Credibility(node)
honesty = Compute Honesty(node)
availability = Compute availability(node)
trust_degree = (credibility + availability + honesty)/3
trust_degree = credibility_weight * credibility + honesty_weight * honesty + availability_weight * availability
Set the default_trust == 0.5
Train the trust management model by the SVM model.
if $trust\_degree \geq default\_trust\_value$ **then**
    $predict$ "$trustworthy$"
else
    $predict$ "$untrustworthy$"
$remove$"$trustworthiness$"

Algorithm 2 takes a node in the network as input and computes its trust degree based on three parameters—credibility, honesty, and availability. The trust degree is then compared to a default threshold value of 0.5 to decide whether the node is trustworthy or not. The algorithm also involves training a trust management model using an SVM classifier to aid in the decision-making process. The algorithm begins by initializing the input node and the output trust degree. It then proceeds to compute the credibility, honesty, and availability of the node. The credibility parameter measures the accuracy of the information provided by the node, the honesty parameter measures the likelihood of the node providing truthful information, and the availability parameter measures the willingness of the node to participate in the network.

Once the three parameters are computed, the trust degree is calculated as the weighted sum of these parameters, with the weights assigned to each parameter based on their importance in the trust evaluation process. The algorithm then compares the computed trust degree to a default threshold value of 0.5, which is used to differentiate trustworthy nodes from untrustworthy ones. The trust management model is trained using an SVM classifier, which takes as input the computed trust degree and outputs a prediction of whether the node is trustworthy or not. The trained model is used to aid in the decision-making process, as it helps to reduce the possibility of false positives or false negatives in identifying trustworthy nodes. Finally, the algorithm removes the trustworthiness attribute, as it is not needed once the decision has been made. The output of the algorithm is the trust degree of the node, which is used to determine whether the node should be allocated a parking space or not. Algorithm 2 provides a simple and effective way to evaluate the trustworthiness of nodes in the network and make informed decisions about parking allocation based on this evaluation.

3.2. Trust Computational Parameter and Formulation

The trust computations in the parking system between the parking stand, RSU, vehicles, and base station are formulated in this section. The trust is established by using credibility (Cr), honesty (H), and availability (Co) as trust parameters. These parameters play a crucial role in determining the trust degree between different nodes. The credibility between the nodes is computed using the formula provided in Equation (1).

$$Cr = \frac{\alpha}{\beta} * 100 \tag{1}$$

Honesty is a metric for evaluating the reliability of a node based on its compliance with the rules and regulations of the smart parking system. Equation (1) calculates the honesty of a node by taking into account the number of successful data packet deliveries (α) and the total number of data packet deliveries (β). One way to assess a node's honesty is to keep track of the number of instances where it violates the system's rules and regulations, such as attempting to park in a restricted area or accessing a parking space without paying the fee. An example mathematical equation to compute the honesty of a node is shown below:

$$H = \frac{\sigma}{\Phi} * 100 \tag{2}$$

In Equation (2), the honesty of a node is calculated by dividing the number of rule violations (σ) by the total number of parking transactions (Φ) and multiplying the result by 100. This results in a percentage value for honesty, with a lower value indicating higher honesty. The availability of a node is determined by subtracting the result obtained from dividing the number of times the node has received assistance by the total number of parking transactions and multiplying the result by 100 from the result obtained from dividing the number of times the node has assisted others by the total number of parking transactions and multiplying the result by 100. This results in a percentage value for availability, with a higher value indicating higher availability.

The equation for calculating the availability of a node is shown in Equation (3). To implement this equation, you need to define a function that computes availability (node).

This function should take in a node as an input and return its availability value. The function must keep track of the number of times the node has provided assistance and received assistance, as well as the total number of parking transactions. By using the above equation, the function can then calculate the availability of the node.

$$Co = \frac{\varphi}{\lambda} - \frac{\tau}{\lambda} \tag{3}$$

In Equation (3), φ represents the total number of times a node has assisted others, λ represents the total number of parking transactions, and τ represents the number of times the node has received assistance. For example, if a node has assisted others 10 times out of a total of 100 parking transactions, and has received assistance 5 times out of a total of 100 parking transactions, its availability could be calculated as follows:

$$C = (10/100) \times 100 - (5/100) \times 100 = 10 - 5 = 5$$

This node would have an availability value of 5. The trust degree between different nodes is computed using these formulas, which helps to maintain trustworthiness.

$$trust\_degree = cr\omega * cr + h\omega * h + co\omega * co \tag{4}$$

The trust degree between the different nodes of vehicles, RSU, parking stand, and base station is computed using Equation (5). The credibility weight is represented by $cr\omega$, where cr is credibility. The honesty weight is represented by $h\omega$, where h is honesty. The availability weight is represented by $co\omega$, where co is availability. The computed trust parameters are stored in the priority table. The computation of the trust degree is an event-driven process that occurs when nodes send data. The algorithm checks the trust degree of nodes and after the computation, the average of credibility, honesty, and availability is taken as the trust degree.

$$Average\_trust = \frac{(credibility + availability + honesty)}{3} \tag{5}$$

Equation (5) calculates the average trust between the different nodes in the parking system, including vehicles, RSUs, parking stands, and the base station. This average trust value can then be used in a support vector machine (SVM) model to distinguish between trustworthy and untrustworthy nodes. The SVM maps the trust value to a hyperplane and labels it as either positive (+1) if the value is equal to or greater than 0.5, or negative (-1) if the value is less than 0.5. In this way, the SVM model is able to evaluate the degree of trust in the parking system. The SVM approach trains the trust parameters to predict the trustworthiness or non-trustworthiness of the data contained in the system, as described in detail in Equation (6).

$$P = \{(Z_1, y_1), (Z_2, y_2), \ldots\ldots, (Z_n, y_n)\} \tag{6}$$

In Equation (6), P represents the training set, and Z_i represents the samples of average trust values, where y_i is the class label. The $y_i \in +1, -1$ value of y_i is either 1 or 0 indicating whether the trust value belongs to a malicious or non-malicious class. If the trust values are transformed linearly in the SVM plane, they are separated linearly; otherwise, they are separated non-linearly. The linear separation of the average trust values draws two lines between the hyperplane, which are classified as a vector of class +1 and a vector of class -1. Many lines can be drawn, and the lines that separate the average trust values equal to or greater than 0.5 and less than 0.5 are called the optimal classification lines. The formula for the hyperplane in the optimal classification line is given as $\omega.Z_i + b = 0$ ($\omega \in \mathbb{R}, b \in \mathbb{R}$), where ω represents the weight of the average trust.

$$\omega.Z_i + b > 0 \tag{7}$$

Equation (7) represents the hyperplane of the SVM model, which is used to classify the desired label class on the hyperplane. This equation is used to predict malicious nodes among the vehicles, parking stands, RSUs, and the base station. The trustworthy nodes are scattered on the positive side of the hyperplane.

$$\omega.Z_i + b < 0 \tag{8}$$

In Equation (8), the hyperplane of the SVM model is identified. It classifies the below-scattered values as the desired label class on the hyperplane. The equation expresses the adjustment of the average trust value to make the edge of the hyperplane of the SVM model. The negative side of the hyperplane is adjusted by the formula illustrated in Equation (10).

$$Hyp_1 : \omega.Z_i + b \geq 1, \qquad for \quad y_1 = 1 \tag{9}$$

$$Hyp_1 : \omega.Z_i + b < 1, \qquad for \quad y_1 = -1 \tag{10}$$

The training set shows projection on Hyp_2 and Hyp_2 are evaluated by vectors and the equal sign is formed by the equation. The maximum edge is measured by $2/\|\omega\|$. Finding that the maximum value of hyperplane and margin of SVM as $2/\|\omega\|$ is equal to calculating the minimum value of $\|\omega\|$. When applied to n-dimensional space, finding the best hyperplane using the SVM is equal to resolving the restricted optimization problem.

$$min_{\omega b} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n} \xi_i \tag{11}$$

$$y_i(\omega.Z_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1, 2, 3, \ldots\ldots, n, \tag{12}$$

A value of C greater than zero denotes the trust parameter, denoting how much attention should be paid to outliers, and ξ_i represents the degree of outliers in the relaxation variable [30].

3.3. SVM Model

The pseudo-code for this classification process is elaborated in Algorithm 3. The algorithm is an SVM-based approach for detecting malicious and compromised nodes in a network using trust parameter values as input features. The algorithm first initializes the SVM model with the trust parameters as input features and the labels as malicious or compromised. Then, the data are split into training and testing sets. The SVM model is trained on the training set, and the performance is evaluated on the testing set using metrics such as accuracy, precision, and recall. Finally, if the performance of the SVM model is satisfactory, it can be used to classify new vehicle nodes as malicious or compromised.

The proposed approach improves the algorithm by adding equations that explain the SVM model in more detail. Specifically, the trust parameter values for node i are denoted by $x_i \in \mathbb{R}^n$, and the label indicating whether node i is malicious or not is denoted by $y_i \in -1, 1$. The SVM model can be represented using a kernel function $K(x_i, x)$ that measures the similarity between x_i and x, and the Lagrange multipliers $\alpha \in \mathbb{R}^m$, weight vector $w \in \mathbb{R}^n$, and bias $b \in \mathbb{R}$ can be used to calculate the SVM model output $f(x)$ as (13).

The performance of the SVM model is evaluated using three metrics: accuracy, precision, and recall. These metrics can be calculated using the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), and the results are discussed in Section 4.

Algorithm 3 The Support Vector Machine Model for Detecting Malicious and Compromised Nodes

Input: Trust parameter values
Output: Detect malicious nodes.

1: **Procedure**
2: Initialize the SVM model with the trust parameters as input features and the labels as malicious or compromised.
3: Let $x_i \in \mathbb{R}^n$ be the trust parameter values for node i and $y_i \in -1, 1$ be the label indicating whether node i is malicious or not.
4: Split the data into training and testing sets.
5: Let X_{train} and y_{train} be the training set of input features and labels, respectively, and X_{test} and y_{test} be the testing set of input features and labels, respectively.
6: Train the SVM model on the training set.
7: Let $\alpha \in \mathbb{R}^m$ be the Lagrange multipliers, and $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ be the weight vector and bias, respectively. The SVM model can be represented as:

$$f(x) = sign(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b) \tag{13}$$

where $K(x_i, x)$ is the kernel function that measures the similarity between x_i and x.
8: Test the SVM model on the testing set.
9: Let y_{pred} be the predicted labels for the testing set.
10: Evaluate the performance of the SVM model using metrics such as accuracy, precision, and recall.
11: Let TP, FP, TN, and FN be the number of true positives, false positives, true negatives, and false negatives, respectively. The metrics can be calculated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{14}$$

$$Precision = \frac{TP}{TP + FP}, \tag{15}$$

$$Recall = \frac{TP}{TP + FN}. \tag{16}$$

12: If the performance of the SVM model is satisfactory, use it to classify new vehicle nodes as malicious or compromised.

Artificial Neural Network Model

In our proposed approach, the dataset used to train the trust management value is processed through an artificial neural network (ANN) algorithm. This is in contrast to the conventional machine learning techniques such as decision tree, random forest, and ANN. The ANN algorithm is defined in Equation (17) as described in the research paper by Kumar et al. [31].

$$Z(x) = \sum_{i=0}^{n} x_i * \omega_i \tag{17}$$

In the proposed approach, the dataset consisting of trust management values is trained using the artificial neural network (ANN) algorithm. This is in contrast to the existing machine-learning approaches, such as decision tree and random forest. Equation (17) of the ANN, as described in [31], is given below. The equation shows how the neural network functions by transforming the input into hidden layers to produce the output. The input, represented by x_i, is processed through the network along with the weight of the parameters, represented by ω. The result of the supervised learning performed by the model is determined by how the network was trained. The predicted value of the ANN

model ranges from 0.0 to 1.0. Values less than 0.4 are considered low trust, while values greater than 0.8 are considered high trust between vehicles, RSUs, parking stands, and the base station. The activation function is used to identify malicious nodes. The formula for the ANN is described in [32].

$$f(z) = \frac{1}{1 + e^{-z}} \tag{18}$$

The activation function consists of e^{-z} is a sigmoid function, it is responsible for the detection of the malicious node. The z is the total trust management input given to ANN. After using the trust management approach, we apply machine learning algorithms to it and check the best performance models. The following mathematical equation represents the computation of trust parameters for a vehicle node in a smart parking system using the artificial neural network algorithm. Credibility is a measure of the reliability and trustworthiness of a vehicle node. It is computed using the following equation:

$$Credibility(n) = \sum_{0.0}^{1.0} w * Cr(n) \tag{19}$$

where n is the vehicle node, w is the weight of the feature, and $Cr(n)$ is the feature vector of the node. Honesty is a measure of the honesty and integrity of a vehicle node. It is computed using the following equation:

$$Honesty(n) = \sum_{0.0}^{1.0} w * h(n) \tag{20}$$

where n is the vehicle node, w is the weight of the feature, and $h(n)$ is the feature vector of the node. availability is a measure of how many nodes are cooperative with a vehicle node. It is computed using the following equation:

$$availability(n) = \sum_{0.0}^{1.0} w * Co(n) \tag{21}$$

In the proposed approach, the dataset consists of trust management values, which are trained using an artificial neural network (ANN) algorithm. The ANN algorithm is used to classify the vehicle nodes as malicious or trustworthy. The equation for the ANN is given in Equation (17), where n represents the vehicle node, w represents the weight of the feature, and $Cr(n)$ represents the feature vector of the node. Once the trust parameters have been computed, they can be used to classify a vehicle node as malicious or trustworthy using the ANN algorithm. The ANN classifier separates the values on the hyperplane and makes a decision based on the average trust value. If the average trust value is greater than or equal to 0.5, the node is considered trustworthy and allowed to transmit data. If the average trust value is less than 0.5, the node is considered untrustworthy and eliminated, not allowing it to transmit data to the base station or any other vehicle.

$$\begin{cases} Trustworthy & \text{if } Average\_trust \geq default\_trust \\ Untrustworthy & \text{if } Average\_trust < default\_trust \end{cases}$$

4. Simulation Outcome

The trust-based parking system was simulated using a Jupiter Notebook. The accuracy of the system was evaluated based on its ability to accurately handle tasks and defend against potential attacks. The trust management approach calculates the number of malicious nodes, while machine learning models identify these malicious nodes. The simulation was run on a core-i9 system with a 1 TB SSD and 32 GB of RAM. The Jupiter Notebook was used to obtain the results of the simulation, utilizing the Keras and TensorFlow libraries, and the complete simulation setup and parameters are illustrated in Table 2.

Table 2. Simulation Parameters for Smart Parking System.

Simulation Parameter	Description	Value
Area of network	Area covered by the simulation scenario	2 square km
Number of nodes	Total number of nodes in the simulation scenario (vehicles and RSUs)	510
Simulation time	Duration of the simulation scenario	24 h
Transmission range	Maximum distance that a node can transmit data to other nodes	300 m
Routing protocol	Protocol used for routing data between nodes	AODV
MAC	Medium access control protocol used in the simulation scenario	IEEE 802.11p
Mobility model	Model used for simulating the movement of vehicles in the scenario	Manhattan mobility model
Transmission rate	Data transmission rate between nodes	6 Mbps
Size of packet	Average size of data packet transmitted between nodes	1 KB
Position of RSU	Location of RSUs in the scenario	Randomly distributed
Average speed of node	Average speed of vehicles in the scenario	30 km/h
Parking stands	Total number of available parking spaces in the parking lot	1000
Traffic density	Density of the traffic on the road network	High
Trust value	A value between 0.0 and 1.0 representing the trust level of nodes in the network	0.0–1.0

Figure 2 shows a correlation graph of the trust parameters used in our simulation, including availability (Co_ob), Honesty (H_ob), Credibility (Cr_ob), and Average_Trust. The correlation graph provides a visual representation of the relationships between the variables. The diagonal value for each parameter is 0.1, indicating a weak positive correlation between each parameter and itself. The other values in the graph represent the strength and direction of the correlation between each pair of parameters. This graph provides a visual representation of the relationships between the trust parameters used in our simulation.

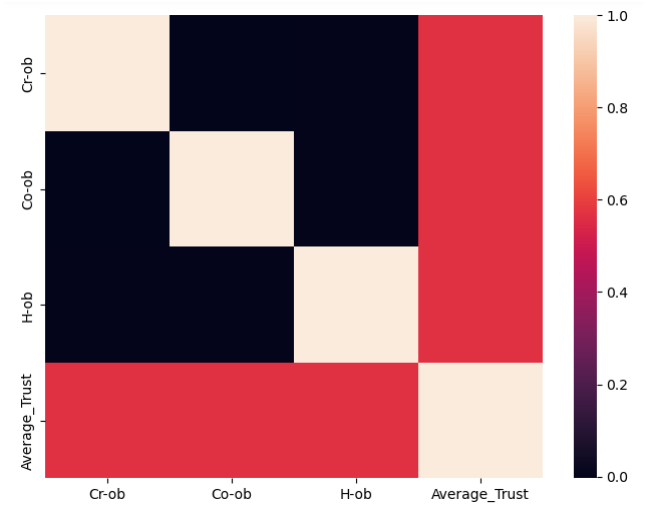

Figure 2. Correlation Feature of Trust-Based Parking System.

In the proposed approach, the dataset used for the trust-based parking system contains 7000 rows of trust values. The decision regarding the trustworthiness of a node is made by an SVM classifier. The SVM classifier separates the data into two classes: trustworthy and non-trustworthy. The values that are equal to or greater than 0.5 are considered trustworthy and plotted on the positive side (+1) of the hyperplane. Conversely, values less than 0.5 are considered non-trustworthy and plotted on the negative side (−1) of the hyperplane.

4.1. Training and Testing of SVM

In this proposed approach, the SVM is used to predict the behavior of nodes. The hyperplane is utilized to separate the data into different classes. The data points on one side of the hyperplane are classified as one class, while the data points on the other side are classified as a different class. The SVM algorithm attempts to find a hyperplane that maximally separates the positive and negative points, where positive points are labeled as trustworthy and negative points are labeled as non-trustworthy. The points closest to the hyperplane are referred to as support vectors and play a crucial role in determining the position of the hyperplane.

Figure 3 shows how to divide nodes into two classes—trusted and untrusted—using the support vector machine (SVM) algorithm. The decision boundary separating these two classes is represented by the hyperplane, with the positive side representing trustworthy nodes and the negative side representing unreliable nodes. The SVM algorithm's objective is to accurately identify the hyperplane that divides these nodes. By locating the hyperplane that maximizes the margin between the two classes of nodes, the SVM algorithm employs a method known as maximum margin classification to accomplish this. The hyperplane and the nearest data points from each class are separated by this margin. The SVM algorithm can produce a decision boundary that is reliable and generalizable by maximizing this margin, making it possible for it to accurately categorize new, unobserved data points as trustworthy or untrustworthy. According to reports, 96.43 percent of the trust management model integrated with the SVM model is accurate. According to the hyperplane produced by the SVM algorithm, the model can correctly categorize 96.43 percent of the nodes as either trustworthy or untrustworthy. With such a high level of accuracy, the model can be applied to a number of fields including social network analysis, e-commerce, and online security. It also shows that the model is efficient at differentiating between trustworthy and untrustworthy nodes.

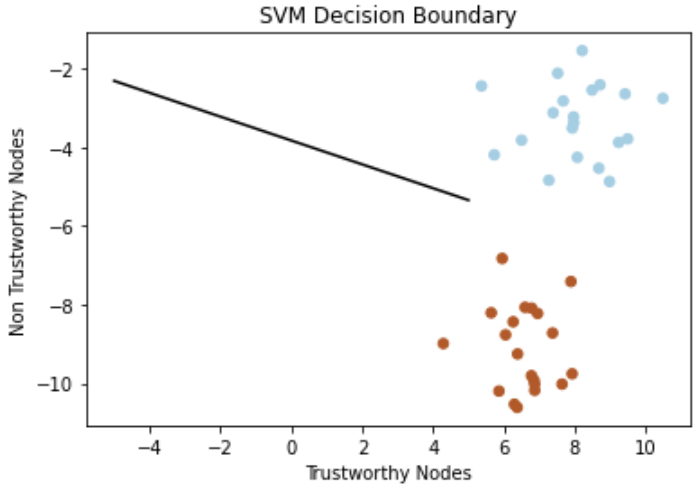

Figure 3. SVM separation of malicious and non-malicious nodes.

4.2. Accuracy Parameter for SVM Model

The proposed methodology's effectiveness is evaluated using precision, recall, and F1-score metrics. These metrics are used to assess the performance of the SVM classifier. Precision is a measure of how many true positive predictions (correctly identified malicious nodes) were made out of all positive predictions made by the model. For example, a precision score of 0.95 indicates that 95% of the positive predictions made by the model are accurate. Recall is a measure of the proportion of true positive predictions made by the model out of all actual positive instances (malicious nodes) in the data.

A recall score of 0.959 means that the model correctly identified 95.9% of the malicious nodes in the data. F1-score is the harmonic mean of precision and recall, and it provides

a single score that reflects the balance between precision and recall. An F1-score of 0.955 indicates that the precision and recall values are similar, and the model has a good balance of both. In the context of malicious and non-malicious node classification, these values suggest that the SVM model has a good performance, with a precision of 0.95, a recall of 0.959, and an F1-score of 0.955.

4.3. Training and Testing of Deep Neural Network

In the training and testing of deep neural networks (DNNs) for detecting malicious nodes in a network, a trust management approach is utilized. This approach involves training the DNN model using trust management features, such as credibility, honesty, and availability, which reflect the behavior, history, and relationships of nodes in the network. The training process involves splitting the labeled data into a training set and a testing set and providing the DNN with the training data to learn the patterns and relationships between the features and the label of malicious or non-malicious nodes.

The DNN model's performance is assessed using the testing data after it has been trained. The model is given fresh, previously unobserved data in this step, and its predictions are contrasted with the actual labels. Metrics such as accuracy, precision, recall, and F1-score, which measure the model's capacity to precisely identify malicious nodes, are used to evaluate the model's performance. The proposed DNN model achieved an accuracy of approximately 90.9%, as shown in Figure 4 whereas the detailed simulation outcome is illustrated by Table 3. During testing or validation, the accuracy fluctuates between 0.89 and 0.909. We trained our smart parking trust management value using a batch size of 32 and an approach value of 1500. Additionally, Figure 5 illustrates that the training loss reduces from 0.15 to 0.14, which is a negligible low value.

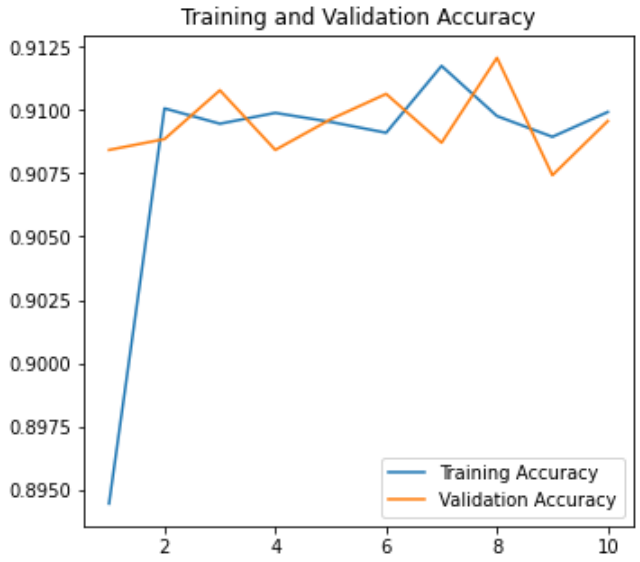

Figure 4. Deep neural network accuracy with the proposed approach.

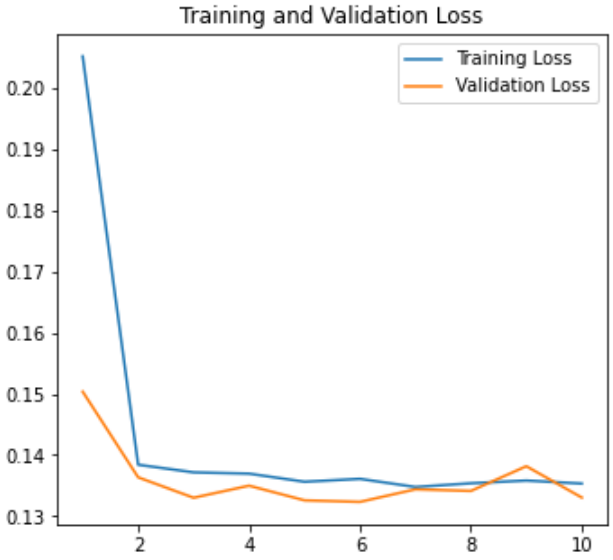

Figure 5. Model loss during training trust management value.

Table 3. Deep Neural Network Accuracy Matrices.

	Precision	Recall	F1-score	Support
0	0.97	0.90	0.93	10,033
1	0.79	0.92	0.85	3967
accuracy			0.91	14,000
macro avg	0.88	0.91	0.89	14,000
weight avg	0.92	0.91	0.91	14,000

4.4. Comparative Analysis

We compared various methods for spotting malicious nodes in a smart parking system, and the results showed that the SVM model performed 96.4 percent better than the DNN model, which scored 90 percent. This suggests that, in terms of accuracy, the SVM model is better suited for this task. It is crucial to remember, though, that accuracy is not the only aspect to take into account when choosing a machine-learning model for a particular issue. Other elements such as interpretability, scalability, and computational complexity should also be considered. We have provided more information on these factors in the bar graph displayed in Figure 6.

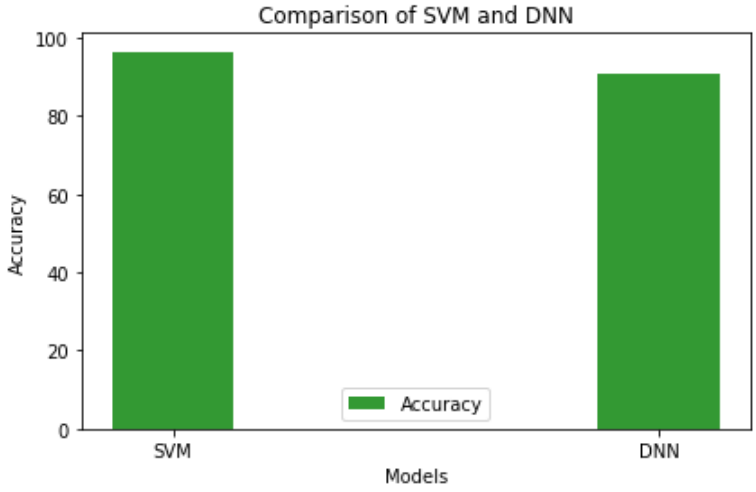

Figure 6. Comparison Analysis of SVM and DNN.

5. Conclusions

In this paper, we presented a hybrid approach to enhance the security and privacy of IoT-enabled smart parking in sustainable smart cities. Our approach combines trust management and machine learning techniques, specifically support vector machine (SVM) and artificial neural network (ANN) models, to detect and eliminate malicious nodes in smart parking. The trust management parameters we used include credibility, availability, and honesty. After evaluating both models, we found that the hybrid SVM classifier with a trust parameters approach outperformed the other models, achieving an accuracy of 96.43% in detecting and eliminating malicious or compromised nodes. On the other hand, the hybrid trust management deep neural network (DNN) model achieved an accuracy of 90.9%. Our results demonstrate the potential of the SVM approach in improving the efficiency and sustainability of urban environments by optimizing the use of parking resources and ensuring responsible data collection and usage. The current study utilizes SVM and ANN models for detecting malicious nodes. In the future, the proposed mechanism could be extended to optimize the computational process, with the aim of reducing energy consumption.

Author Contributions: Methodology, J.A. and M.F.K.; Validation, M.F.K.; Investigation, J.A.; Writing–original draft, M.F.K.; Writing–review & editing, J.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tang, C.S.; Chiang, Y.K.; Tsou, Y.D.; Ju, J.H.; Twu, C.Y. A converged network architecture for ICT and IoT combined applications. In Proceedings of the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, Japan, 5–7 October 2016; pp. 1–4.
2. Awan, K.A.; Din, I.U.; Almogren, A.; Rodrigues, J.J. AutoTrust: A privacy-enhanced trust-based intrusion detection approach for internet of smart things. *Future Gener. Comput. Syst.* **2022**, *137*, 288–301. [CrossRef]
3. Wang, S.; Wang, S.; Liu, Z.; Zhang, Q. A role distinguishing Bert model for medical dialogue system in sustainable smart city. *Sustain. Energy Technol. Assessments* **2023**, *55*, 102896. [CrossRef]
4. Arif, S.; Taweekun, J.; Ali, H.M.; Noranai, Z.; Qadeer, A. Development of a cost effective approach toward energy efficient buildings by design, fabrication and economical analysis of air conditioning pods: A case study of a bus station in Thailand. *Case Stud. Therm. Eng.* **2023**, *41*, 102534. [CrossRef]
5. Oad, A.; Ahmad, H.G.; Talpur, M.S.H.; Zhao, C.; Pervez, A. Green smart grid predictive analysis to integrate sustainable energy of emerging V2G in smart city technologies. *Optik* **2023**, *272*, 170146. [CrossRef]
6. Rani, P.; Sharma, R. Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. *Comput. Electr. Eng.* **2023**, *105*, 108543. [CrossRef]
7. George, A. Distributed Messaging System for the IoT Edge. Ph.D. Thesis, The University of North Carolina at Charlotte, Charlotte, NC, USA, 2020.
8. Kumar, P.; Lobine, D. Re-assessing urban sustainability in the digital age: A new SWOT methodology for cities. In *Resilient and Sustainable Cities*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 193–225.
9. Upadhyay, D.; Agarwal, A. Smart Bin: An Approach to Design Smart Waste Management for Smart Cities. In *Revolutionizing Industrial Automation through the Convergence of Artificial Intelligence and the Internet of Things*; IGI Global : Hershey, PA, USA, 2023; pp. 177–200.
10. Miyasawa, A.; Akira, S.; Fujimoto, Y.; Hayashi, Y. Forecast of area-scale behaviours of behind-the-metre solar power and load based on smart-metering net demand data. *IET Smart Cities* **2023** , *5*, 19–34. [CrossRef]
11. Pal, M.S.; Bhatia, M. Smart Solid Waste Management System Using IoT Technology: Comparative Analysis, Gaps, and Challenges. In *Intelligent Cyber Physical Systems and Internet of Things: ICoICI 2022*; Springer: Cham, Switzerland, 2023 ; pp. 795–811.
12. Cuzzocrea, A. Privacy and security of big data: Current challenges and future research perspectives. In Proceedings of the First International Workshop on Privacy and Secuirty of Big Data, Shanghai, China, 7 November 2014 ; pp. 45–47.

13. Alsafery, W.; Alturki, B.; Reiff-Marganiec, S.; Jambi, K. Smart car parking system solution for the internet of things in smart cities. In Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–5.
14. Awan, K.A.; Din, I.U.; Almogren, A. A Blockchain-Assisted Trusted Clustering Mechanism for IoT-Enabled Smart Transportation System. *Sustainability* **2022**, *14*, 14889. [CrossRef]
15. Nikhar, M.; Kamath, S. IoT-Based E-Parking System for Multiplexes and Shopping Malls. In *Proceedings of the Fourth International Conference on Communication, Computing and Electronics Systems: ICCCES 2022*; Springer: Cham, Switzerland, 2023; pp. 81–89.
16. Sheng, N.W.; Wan Muda, W.M.; Annuar, A.Z.; Wan Hassan, W.H. Parking System Using Radio-Frequency Identification (RFID) Technology. In *Fundamental and Applied Sciences in Asia: International Conference on Science Technology and Social Sciences (ICSTSS 2018)*; Springer: Cham, Switzerland , 2023; pp. 81–90.
17. Aditya, A.; Anwarul, S.; Tanwar, R.; Koneru, S.K.V. An IoT assisted Intelligent Parking System (IPS) for Smart Cities. *Procedia Comput. Sci.* **2023**, *218*, 1045–1054. [CrossRef]
18. Shahzad, A.; Gherbi, A.; Zhang, K. Enabling Fog–Blockchain Computing for Autonomous-Vehicle-Parking System: A Solution to Reinforce IoT–Cloud Platform for Future Smart Parking. *Sensors* **2022**, *22*, 4849. [CrossRef] [PubMed]
19. Muqeet, H.A.; Munir, H.M.; Javed, H.; Shahzad, M.; Jamil, M.; Guerrero, J.M. An energy management system of campus microgrids: State-of-the-art and future challenges. *Energies* **2021**, *14*, 6525. [CrossRef]
20. Chatzigiannakis, I.; Vitaletti, A.; Pyrgelis, A. A privacy-preserving smart parking system using an IoT elliptic curve based security platform. *Comput. Commun.* **2016**, *89*, 165–177. [CrossRef]
21. Lookmuang, R.; Nambut, K.; Usanavasin, S. Smart parking using IoT technology. In Proceedings of the 2018 5th International Conference on Business and Industrial research (ICBIR), Bangkok, Thailand, 17–18 May 2018; pp. 1–6.
22. Badr, M.M.; Al Amiri, W.; Fouda, M.M.; Mahmoud, M.M.; Aljohani, A.J.; Alasmary, W. Smart parking system with privacy preservation and reputation management using blockchain. *IEEE Access* **2020**, *8*, 150823–150843. [CrossRef]
23. Venkateswaran, V.; Prakash, N. Intelligent approach for smart car parking reservation and security maintenance system. *IJRET Int. J. Res. Eng. Technol.* **2014**, *3*, 248–251.
24. Said, A.M.; Kamal, A.E.; Afifi, H. An intelligent parking sharing system for green and smart cities based IoT. *Comput. Commun.* **2021**, *172*, 10–18. [CrossRef]
25. Timpner, J.; Schürmann, D.; Wolf, L. Trustworthy parking communities: Helping your neighbor to find a space. *IEEE Trans. Dependable Secur. Comput.* **2015**, *13*, 120–132. [CrossRef]
26. Yan, G.; Olariu, S.; Weigle, M.C.; Abuelela, M. SmartParking: A secure and intelligent parking system using NOTICE. In Proceedings of the 2008 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China, 12–15 October 2008; pp. 569–574.
27. Anderson, E.C.; Okafor, K.C.; Nkwachukwu, O.; Dike, D.O. Real time car parking system: A novel taxonomy for integrated vehicular computing. In Proceedings of the 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, Nigeria, 29–31 October 2017; pp. 1–9.
28. George, A.; Ravindran, A.; Mendieta, M.; Tabkhi, H. Mez: An adaptive messaging system for latency-sensitive multi-camera machine vision at the iot edge. *IEEE Access* **2021**, *9*, 21457–21473. [CrossRef]
29. Awan, K.A.; Din, I.U.; Almogren, A.; Kim, B.S.; Altameem, A. vTrust: An IoT-Enabled Trust-Based Secure Wireless Energy Sharing Mechanism for Vehicular Ad Hoc Networks. *Sensors* **2021**, *21*, 7363. [CrossRef] [PubMed]
30. Ye, J.; Cheng, X.; Zhu, J.; Feng, L.; Song, L. A DDoS attack detection method based on SVM in software defined network. *Secur. Commun. Netw.* **2018**, *2018*, 9804061. [CrossRef]
31. Kumar, M.; Mukherjee, P.; Verma, K.; Verma, S.; Rawat, D.B. Improved deep convolutional neural network based malicious node detection and energy-efficient data transmission in wireless sensor networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *9*, 3272–3281. [CrossRef]
32. Hasan, M.; Islam, M.M.; Zarif, M.I.I.; Hashem, M. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet Things* **2019**, *7*, 100059. [CrossRef]