electronics

Article

Counteracting Selfish Nodes Using Reputation Based System in Mobile Ad Hoc Networks

Muhammad Fayaz [1,2], Gulzar Mehmood [1], Ajab Khan [1], Sohail Abbas [3], Muhammad Fayaz [4] and Jeonghwan Gwak [5,6,7,8,*]

[1] Department of Computer Science and IT, University of Malakand, Chakdara 18800, Khyber Pakhtunkhwa, Pakistan; m.fayaz@qmul.ac.uk (M.F.); gulzar.mahmood@uom.edu.pk (G.M.); ajabkhan@uom.edu.pk (A.K.)
[2] School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK
[3] Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah 27272, United Arab Emirates; sabbas@sharjah.ac.ae
[4] Department of Computer Science, University of Central Asia, Naryn 722918, Kyrgyzstan; muhammad.fayaz@ucentralasia.org
[5] Department of Software, Korea National University of Transportation, Chungju 27469, Korea
[6] Department of Biomedical Engineering, Korea National University of Transportation, Chungju 27469, Korea
[7] Department of AI Robotics Engineering, Korea National University of Transportation, Chungju 27469, Korea
[8] Department of IT and Energy Convergence (BK21 FOUR), Korea National University of Transportation, Chungju 27469, Korea
* Correspondence: jgwak@ut.ac.kr

Abstract: A mobile ad hoc network (MANET) is a group of nodes constituting a network of mobile nodes without predefined and pre-established architecture where mobile nodes can communicate without any dedicated access points or base stations. In MANETs, a node may act as a host as well as a router. Nodes in the network can send and receive packets through intermediate nodes. However, the existence of malicious and selfish nodes in MANETs severely degrades network performance. The identification of such nodes in the network and their isolation from the network is a challenging problem. Therefore, in this paper, a simple reputation-based scheme is proposed which uses the consumption and contribution information for selfish node detection and cooperation enforcement. Nodes failing to cooperate are detached from the network to save resources of other nodes with good reputation. The simulation results show that our proposed scheme outperforms the benchmark scheme in terms of NRL (normalized routing load), PDF (packet delivery fraction), and packet drop in the presence of malicious and selfish attacks. Furthermore, our scheme identifies the selfish nodes quickly and accurately as compared to the benchmark scheme.

Keywords: mobile ad hoc networks; dynamic topology; multipath routing; secure routing; reputation ratio; trust management; selfish node; malicious node

Citation: Fayaz, M.; Mehmood, G.; Khan, A.; Abbas, S.; Fayaz, M.; Gwak, J. Counteracting Selfish Nodes Using Reputation Based System in Mobile Ad Hoc Networks. *Electronics* **2022**, *11*, 185. https://doi.org/10.3390/electronics11020185

Academic Editor: Yosef Pinhasi

Received: 30 October 2021
Accepted: 23 December 2021
Published: 7 January 2022

1. Introduction

A mobile ad hoc network (MANET) can be formed without the existence of centralized controlling authority or access points, every node acting in an autonomous mode, i.e., nodes are free to join or leave the network. Nodes can communicate to distant nodes via the intermediate nodes and can freely change location, but with the characteristic that mobility network topology changes regularly. Because of hardware malfunctions, malicious attacks, and environmental interference, the sensing nodes and wireless links in the network are prone to failure [1,2]. Thus, MANETs require adaptive protocols that link mobile nodes after the changes induced by random topology. Such networks are robust and scalable because of their distributed nature and can be set up anywhere at any time [3]. Because of these features, a MANET is an effective key and can therefore be

used in various applications, such as emergencies (e.g., natural disaster relief scenarios), military operations, vehicular networks, disaster management, underwater network, robot network [4,5], personal area network, etc. [6,7]. These networks play an integral role in the internet of things (IoT) paradigm [8]. Therefore, a MANET is the best option for social internet of things (SIoT) applications, which are characterized by mutual trust and interests [9]. Routing in MANETs depends on intermediary nodes because every node in the network performs the role of host and router [10,11]. More specifically, they can function as sources, destinations, or intermediate routers to transmit and receive data or forward other node packets based on their movements in the network. This cooperative event of forwarding of packets is used to ensure the reliability of time-dependent tasks. If a node is within another node's transmission range, they can directly communicate; otherwise, they will communicate through intermediate nodes, thereby constituting multi-hop routing. This multi-hop communication makes it possible to solve the low range communication of nodes, but it may not be viable without node collaboration or cooperation [12,13].

One of the most important challenges in MANETs is the lack of cooperation in some nodes in multi-hop communications owing to limited resources (e.g., battery power) and mobility [14]. The noncooperative nodes, known as selfish nodes, selfishly drop packets from other nodes to save their resources and utilize network resources for personal gain. On the other hand, malicious nodes intend to harm the network and exploit its resources. As the number of such nodes increases, it severely disrupts network operations [15]. In order to counteract the effects of such nodes, they must be detected and isolated from the network. Different approaches have been proposed for detecting selfish and malicious nodes. Reputation-based schemes are one of the most popular methods to encourage nodes to cooperate for packet forwarding, in which each node earn a reputation value based on feedback from other nodes [16–19]. Selfish nodes are discouraged by penalizing them with low reputation and ultimately isolating them from the network.

1.1. Motivation

To deal with selfishness (or packet dropping misbehavior), cooperation enforcement schemes have been proposed in [20,21]. Among those schemes, reputation-based schemes are considered more promising and scalable due to their distributed nature and suitability for ad hoc networks. However, there are three key problems in the existing reputation-based schemes that motivate us to contribute to this work:

- How to detect malicious nodes that evade the detection process? Nodes that do not participate in route discovery (i.e., dropping route request packets) cannot be included in the path construction process; therefore, they are not recorded in any routing path; as a result, they are not acting as forwarders or contributors, rather always acting as the traffic sources, i.e., the network service consumers. Reputation-based systems monitor the packet forwarding behavior only while these nodes maliciously excluded themselves from the routing paths; hence, they go undetected [14,20].
- How to capture nodes that intentionally select bad locations? Some malicious nodes strategically position themselves in the network, such as at corners or network boundaries, so that they always act as source nodes; in other words, they are never chosen as forwarders [20,22].
- How to prevent nodes from exploiting reputation thresholds? As soon as a node gains substantial reputation value, it starts dropping packets again for a specific time, i.e., nodes can contribute less and consume more without being detected. These sorts of attackers are partial droppers.

Unlike the existing works, where the reputation of a node was only monitored by its neighbors, involving the source node and forwarding nodes in the reputation calculation process and keeping track of the node's consumption can mitigate these problems. The specific contributions of this work are listed below.

1.2. Contributions

This paper aims to exploit the consumption to contribution (C2C) information to detect and isolate selfish nodes in MANETs. Three main contributions of this work follow:

- Novel Reputation-based Framework: a simple reputation-based framework is designed, which uses the C2C ratio of each node in the network for selfish node detection and cooperation enforcement. Nodes failing to cooperate are detached from the network to save resources of other nodes with a good reputation.
- Performance against Attacker Types: the proposed scheme is investigated against two types of attacks, namely, malicious attacks and partial droppers or selfish attacks.
- Performance Evaluation: it is demonstrated that the proposed scheme outperforms the benchmark scheme in terms of NRL (normalized routing load), PDF (packet delivery fraction), and packet drop in the presence of malicious and selfish attacks. Furthermore, the scheme identifies the selfish nodes quickly and accurately as compared to the benchmark scheme. Moreover, it is also shown that the proposed scheme is scalable to large scale networks.

1.3. Paper Organization

The remainder of the paper is organized the following way: Section 2 presents the background and literature review. Section 3 states the proposed reputation-based scheme. The simulation results are presented in Section 4. Finally, the conclusion and future work are illustrated in Section 5.

2. Background and Literature Review

For the isolation of selfish nodes in mobile ad hoc networks, different cooperation enforcement schemes have been presented in the literature. These schemes can be classified as credit-based or reputation-based schemes, functions of both types of schemes are the same i.e., to enforce a node for packet forwarding activity or to isolate noncooperative nodes from the network, but these schemes use different mechanisms to evaluate node behavior in MANETs. A comparison of existing works is given in Table 1 and further explained in the next section.

Table 1. Existing works with pros and cons.

Scheme	Reputation Based	Credit Based	Features	Limitations
Marti et al. [23]	√		Encourage good nodes	No punishing mechanism for selfish nodes
Buchegger et al. [24]	√		Punish selfish nodes	False alarm messages
Michiardi et al. [25]	√		Punish selfish nodes	Second chance mechanism
Jiangyi Hu et al. [26]	√		Local reputation information	False accusation problem
Bansal et al. [27]	√		Direct reputation mechanism	Second chance mechanism
Buttyan et al. [28]		√	Trading among nodes	Wastage of nuglets
Zhong et al. [29]		√	Centralized credit management	Temper proof hardware, fairness problems

2.1. Reputation-Based Schemes

In reputation-based schemes, every node in the network monitors other nodes' packet forwarding activities; if a node forwards packets for other nodes it will be considered as a good node. On the other hand, if a node drops other nodes' packets, it will be considered a selfish or misbehaving node. Thus, a node's reputation will be increased upon the forwarding of packets for other nodes and decreased, otherwise. In such schemes, every node in the network evaluates and shares previous information of communications (experiences) with neighboring nodes and decides in collaboration regarding the behavior of a node, i.e., to establish whether a node is supportive or uncooperative. Reputation-based schemes define three objectives [30,31].

- To provide evidence to distinguish between supportive and uncooperative events;
- To inspire the helpful performance of a node in a network;
- If a node is uncooperative, it must be stopped from consuming services of other nodes, or enforce it for assistance or cooperation.

In these schemes, a node assesses the behavior of another node through local observations by using first-hand experience (i.e., through its direct interactions) or globally (using reports shared by other trustworthy nodes). Following are some of the reputation-based schemes.

Marti et al. [23] proposed a solution that consists of two methods, "watchdog and pathrater", to prevent selfishness and to increase good throughput of nodes. Watchdog is used to identify selfish nodes by overhearing in promiscuous mode, if the next node forwards a packet, then it will be considered as a good behaving node and vice versa; hence, a node rating can be increased or decreased accordingly. The pathrater is then used, which is responsible to constitute or select paths that do not include selfish nodes. There is no mechanism to punish selfish or misbehaving nodes; which is later addressed by the reputation scheme proposed in [24]. Buchegger et al. [24] provide an extension to the DSR protocol named CONFIDANT (cooperation of nodes, fairness in dynamic ad hoc networks), which is a reputation-based scheme. The CONFIDANT uses four components to detect selfish nodes: monitor, reputation system, trust, and path manager. Monitor overhears its neighbor's activities using the watchdog mechanism, discussed above, and informs the trust manager accordingly. For the detection of untrustworthy nodes, trust manager keeps the record of the reputation rating of the nodes in the network. Whenever a misbehaving event occurs, the trust manager broadcasts alarm messages in the network. Moreover, trust manager uses the trust level of the sender nodes to confirm the trustworthiness of alarm messages. The reputation system maintains two lists, one is a rating list and the second is a black list; the rating list contains ratings of each node which is based on first-hand or second-hand information (information received from other nodes). The reputation system then recognizes selfish or malicious nodes based on reputation ratings. If a node rating falls below a predefined threshold, the reputation system then informs the path manager. The path manager decides to include or exclude a node from the path depending on the reputation rating of that node. Thus path manager selects a route with no selfish nodes and also rejects traffic from misbehaving nodes. In this scheme, the alarm messages has the disadvantage of rumors spreading, due to which the attacker node can spread false alarm messages indicating certain node(s) as being selfish. In this scheme, the alarm messages have the disadvantage of rumor spreading [32], due to which the attacker node can spread false alarm messages indicating certain node(s) as being selfish. The authors provided a solution to this problem in their work given in [32]. CORE (collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks) [25] is a reputation-based scheme that works on top of DSR protocol. This considers the MANET as a society to remain trusted and where nodes must be cooperative. CORE uses the following three types of reputation information:

- Reputation considered through direct observations;
- Reputation information is received from one node concerning others;
- Functional reputations related to a certain function and the weight of the function depends on the priority or status of the function.

At last, collective reputation is calculated based on these three sources of reputation information. Hence, a node with poor reputation value will be secluded from the network. However, if a node increases its contribution to the network, then the node may rejoin the network.

Jiangyi Hu et al. [26] proposed LARS (locally aware reputation system for mobile ad hoc networks). LARS uses local reputation information in which every node in the network monitors its one-hop neighbors for packet forwarding activities and calculates node reputation accordingly. Hence, a node having a low reputation value will be declared as selfish and its neighbors are notified through a warning message. To overcome the false accusation problem, the warning message must be received from multiple nodes to verify

the warning message. OCEAN (observation-based cooperation enforcement in ad hoc networks) [27] considers only direct observations to decide a node's behavior. OCEAN uses five components to detect a selfish node, which are given below.

- NeighborWatch: is used to observe neighboring nodes using the watchdog mechanism to determine whether or not a node forwards packets for others;
- RouteRanker: maintains a rating record of adjacent nodes. Initially, a neutral rating is assigned to all the nodes and then increases or decreases according to the receipt of positive or negative events, respectively, received from the NeighborWatch component;
- Rank-Based Routing: this component is responsible for selecting a path having no selfish nodes according to the information generated by the NeighborWatch component;
- Malicious Traffic Rejection: rejects all packets from selfish nodes to stop service provision to selfish nodes from other nodes;
- Second Chance Mechanism: this component periodically gives a second chance to nodes previously declared as selfish. The second chance mechanism is based on the assumption that the node forwards packets, but due to some technical problems or a bad environment the node was unable to maintain its reputation ratings, for example, accidental link errors.

In [15], a central intelligent technique called SDA (separation of detection authority) is presented to consider network reliability. By utilizing three watchdogs, a payment punishment scheme (PPS) was proposed to send messages to the neighbor nodes, monitor them, and report on their behavior [33]. The method clusters nodes, and the cluster head uses a modified extended Dempster–Shafer model, watching out for selfish nodes with the use of watchdogs. Based on the Dempster–Shafer evidence theory, a trust model approach is proposed in [13], which is based on direct trust between nodes and indirect trust between neighboring nodes.

2.2. Credit Based Schemes

Cooperative nodes in credit-based schemes are rewarded in terms of virtual currency [34]. For packet forwarding activity, the source or destination node must pay virtual currency to intermediate nodes. Following are examples of credit-based schemes. Nuglets [28] was the leading perception used to prevent the selfish behavior of nodes in MANETs. The source or destination nodes pay in terms of virtual currency, called nuglets, to the intermediate nodes for their packet forwarding services. For charging the nodes for packet forwarding services, two models are used: packet purse model (PPM) and packet trade model (PTM). In PPM, the source node must pay the number of nuglets to the intermediate nodes for their packet forwarding services. The sender must provide a sufficient number of nuglets with the packets to the intermediate nodes as service charges for their forwarding activity. The issue with this model is that, if more or fewer nuglets are loaded than required, extra nuglets will be lost and if less, then the packet may not reach the destination. Another problem with this model is that if link failures or technical faults occur in the middle of the transaction, then the nuglets charged by some nodes will be wasted [18]. On the other hand, in the packet trade model, every forwarding node in the route purchases the packets from the preceding one and sells the packet to the next node along the path; finally, the destination node purchases the packet. The author in [19] provides the credit-counter concept to enhance PTM. A node counter will be decreased upon sending their own packets while it will increase on forwarding packets for other nodes in the network. To persist in the network, a node must balance its sending and forwarding activities. In Sprite [29], the credit administration relies on a central authority known as credit clearance service (CCS). For charging packet forwarding activity, a node submits a receipt for credit to the CCS. The CCS provides credit to the node after confirmation of the receipt from the next node. The CCS defines the rate for sending and forwarding the packets; thus, the sender loses the credit while intermediate nodes receive credit if CCS obtains satisfactory verification from the next node. Some other credit-based schemes are presented in [35–38]. Credit-based schemes provide a flexible way for nodes to

manage credit, but these schemes did not receive wide acceptance because of the following shortcomings [39].

- Increased Cost: to secure credit management, these schemes require tamper-proof hardware for every node in the network, which may increase cost;
- Lack of Fairness: there is an unequal chance available to nodes for earning credit. Nodes at the middle of the network have a high chance of earning credit while nodes located at the boundaries have less or no chance to obtain credit;
- High Overhead: payment give-and-take may increase overhead.

In the existing schemes, reputation is a measure of cooperation computed by each node for others. This does not indicate anything about the services consumed by nodes. Therefore, such schemes have limited performance against selfish and malicious nodes. In particular, nodes that do not participate in the route establishment process (i.e., nodes drop route request and route reply messages) make use of network resources to their own advantage [40]. Moreover, nodes that locate at positions where they are always treated as source nodes freely enjoy the benefits of other nodes, but without contributing the same back to the network [41,42]. Similarly, detecting and isolating partial droppers is also challenging since these nodes take advantage of the reputation threshold [43]. The selfish and malicious nodes discussed in the cases above act as free riders, where they usually consume network services without providing any services back to the network.

3. Proposed Reputation-Based Scheme

In this section, the proposed reputation-based scheme is presented while focusing on its different components along with their functionalities for selfish node detection.

3.1. Schemes Description

Unlike the existing methods, where reputation is a measure of cooperation without considering the resources being utilized by nodes, the C2C ratio, on the other hand, provides the overall behavioral history of nodes, for example, how many resources a node exploited for its own benefits and how much it has contributed to the network. Based on our mechanism, each node in an ad hoc network implements an autonomous reputation evaluation scheme that aims to identify and isolate selfish neighbors.

MANET is a set of mobile nodes $\boldsymbol{N} = \{n_1, n_2, \cdots, N\}$ communicating in an ad hoc manner. All the nodes act as a host to send their own packets and a router to forward other node packets. Packets are assumed to be forwarded hop-by-hop. The communication links are considered bidirectional, and the wireless channels are assumed to be error-free. Each node has an omnidirectional antenna for bidirectional communication [44]. To compute the reputation of nodes, every node in the network maintains a C2C table to record the consumption and contribution of other nodes. The consumption (sent packets) of a node n is the number of packets it sent through other nodes in the network, whereas the contribution is the number of packets node n forwarded for other nodes. Based on C2C information, a node n reputation can be calculated as

$$R_1^n = \frac{\sum_{pkt=1}^{P} F_{pkts}}{\sum_{pkt=1}^{P} F_{pkts} + D_{pkts}} \tag{1}$$

$$R_2^n = \frac{\sum_{pkt=1}^{P} F_{pkts}}{\sum_{pkt=1}^{P} F_{pkts} + S_{pkts}} \tag{2}$$

where the F_{pkts} (forward packets) represents contribution, S_{pkts} (sent packets) is the consumption, and D_{pkts} (drop packets) denote the dropped packets. The R_1^n is the reputation value used to detect the type-I attack (malicious attack) and R_2^n is for the type-II attack

(partial droppers or selfish attack). A node can use network services if the following condition holds

$$R_1^n and\ R_2^n \geq \tau \quad (3)$$

where τ is the tolerable reputation threshold and a node satisfying the above condition is considered as a good node. If a node reputation falls below the threshold, such nodes will be declared as selfish or malicious nodes and secluded from the network.

3.2. Design and Modules

Because of the broadcast medium of MANETs, when a node sends or forwards a packet, it is overheard by all its neighbors [22]. All neighbors in our proposed approach monitor packet sending and forwarding operations and calculate the sender's or forwarder's reputation—based on send and forward events. Our scheme will run on the top of DSR. The monitor observes the packet sending and forwarding activity to compute the C2C. The monitor forwards C2C to the reputation module and path manger ensures avoiding the route through selfish nodes. Finally, selfish node isolation is used to isolate nodes that are below the reputation threshold.

3.2.1. DSR Agent (Route Discovery)

The DSR [28] served as the foundation for our strategy. On the routing level, the DSR protocol's primary function is route-finding and packet transmission. In NS-2, the DSR agent is the fundamental module responsible for routing-related events in the network, as well as packet overhearing capability.

3.2.2. The Monitor (C2C Based Detection Mechanism)

The monitor is the basic component of the proposed scheme, which is responsible for identifying the consumption (send packets), contribution (forward packets for others), and drops packets of each node in the network. Unlike prior methods, in our proposed scheme, the sender, neighbor, and forwarder nodes monitor other nodes in their communication range and calculate the C2C for those nodes in the network. In the following text, three different perspectives from three different nodes viewpoints are discussed while calculating C2C.

- Monitoring by the Data Source:

After a route request (RREQ) and route reply (RREP) process, source node *S* (as shown in Figure 1 sends a packet to destination *D* via the intermediated node *F*. It is worth noting that the difference between *S* and *F* is that *S* is the data owner whereas *F* is not. Reaching data from *S* to *D* will benefit both *S* and *D*; for example, suppose *S* is the file uploader and *D* is the file downloader. However, for *F* it is just a community service consuming its battery for the general good. Before delivering the packet to the next-hop *F*, the sender node *S* copies it into a buffer and sets the timer. Due to the broadcast nature of the wireless networks, if the intermediate node *F* forwards the packet before the timer expires, the sender node will also receive a copy of it (also called passive acknowledgment). The sender updates the contribution of the forwarder node *F* in the C2C table. If the timer expires without a matching packet being received, the sender *S* records a packet drop event for *F*. The monitor then communicates this information to the reputation system for reputation calculation. The process is given in Algorithm 1.

- Monitoring by Forwarders:

After responding to the RREQ of the source node *S*, the intermediate node *F* receives a packet from the sender node. If the sender is a new node, its details are added to the C2C table; otherwise, it is checked in its selfish list. If this node is available in the selfish list, then *F* discards the packet. If the sender is not in the selfish list, it updates the consumption node *S* (because *S* is the data source) and passes it to the reputation module for reputation calculation and threshold checking. Details are given in Algorithm 2.

Algorithm 1. Reputation calculation for intermediate node (*F*) by source node (*S*)

```
1. Initiate route discovery process
2. Create the RREQ packet
3. Broadcast the RREQ
4. If  receive the RREP from the neighbors then
5.     Send packet to the next hope, start the timer, and listen to it
6.     If  overhear the same copy from the next hope then
7.         Update the contribution in the C2C table
8.     Else
9.         Update drop packets
10.    End if
11. Else
12. Go to step 2
13. End if
14. Calculate reputation-based on C2C
15. If  reputation < threshold then
16.    Add next hope to selfish list and go to step 1
17. End if
```

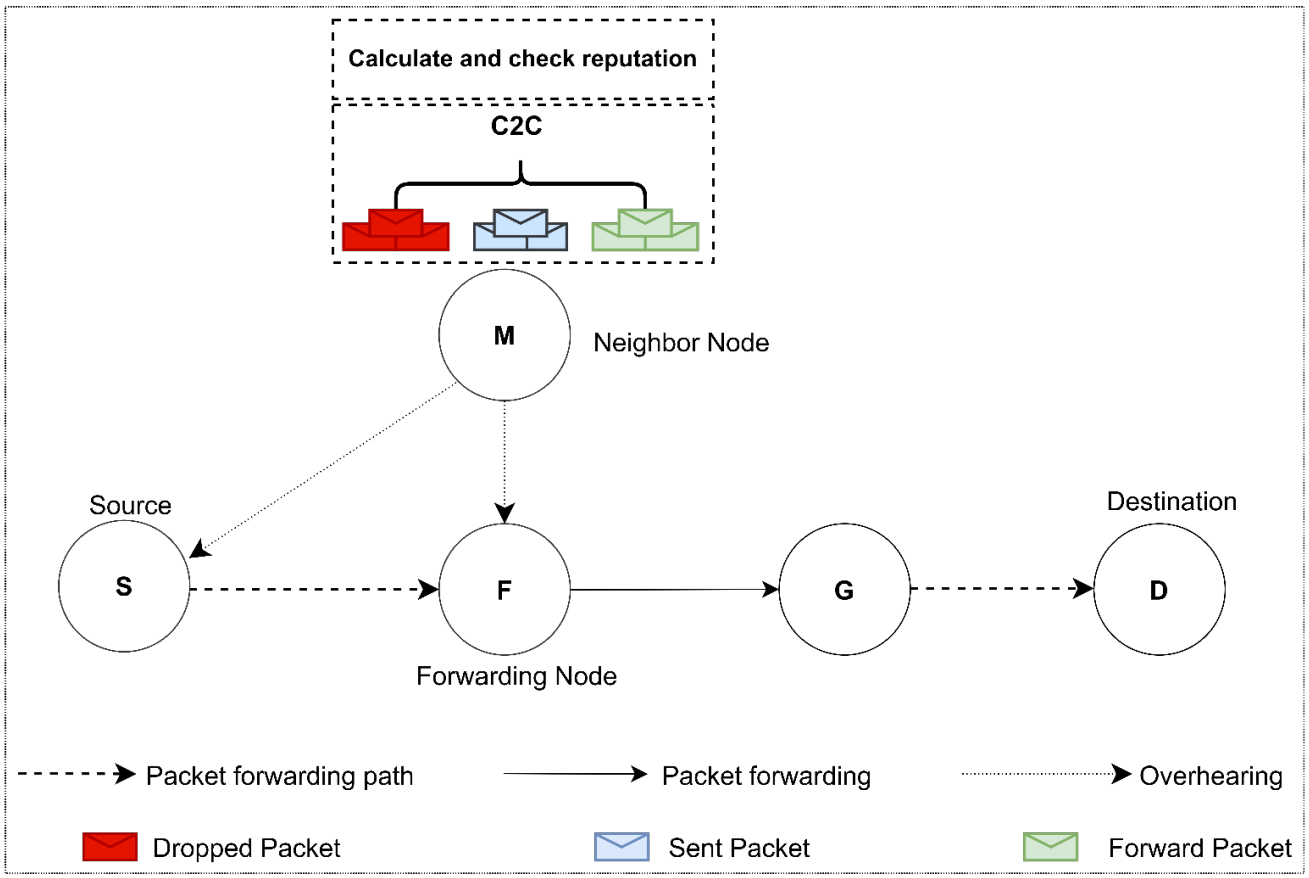

Figure 1. Illustration of the proposed scheme, where nodes monitor the C2C of other nodes and calculate the reputation accordingly.

Algorithm 2. Reputation calculation for sender node (*S*) by intermediate node (*F*)

```
1. Receive packet
2. If   packet type == RREQ then
3.      If   current node == destination then
4. Send RREP
5.      else
6.           Add itself in the path and broadcast the packet
7.         End if
8. Else   packet type == data
9.         If   source == new node then
10.             add its details to the C2C table
11.         Else check it in the selfish nodes list
12.                 If   available in the selfish list then
13.                      Discard the packet
14.                 Else update the consumption
15.                 End if
16.         End if
17.   End if
18. Calculate reputation-based on C2C
19. If reputation < threshold then
20.      Add sender to the selfish list and discard the packet
21. End if
```

- Monitoring by Neighbors:

The neighbor node *M* can overhear the communication between the sender *S* and the forwarder node *F*. It is notable that *M* is a neighbor which only overhears the communication and not acting in the routing path. As the sender node *S* sends the packet to the forwarder *F*, the neighbor node *M* saves the overheard copy of the packet in a cache. In the next step, within a certain time threshold, if *M* overhears a packet from the forwarder node *F,* then *M* compares this overheard packet with the cached one. If both packets are the same, then *M* updates the consumption of the sender node *S* and contribution of the *F* node in its C2C table. If both packets are not the same, *M* updates the consumption of the *F* node. If no packet is overheard from *F*, then *M* assumes that *F* dropped the packet received from *S* and updates the dropped record of the *F* node. Finally, *M* sends C2C information to the reputation module for reputation calculation. The details are given in Algorithm 3.

Algorithm 3. Reputation calculation for sender node (*S*) and intermediate node (*F*) by neighbor node (M)

```
1. If   Overhear a packet from the sender then
2.      Cache a copy of this packet and listen to the forwarder
3.      If   overhear a packet from the forwarder then
4.           compare this packet with the cached one
5.           If   both are the same then
6.                Update consumption of the sender
7.                Contribution of the forwarder
8.           End if
9.      Else
10.          Update the dropped record of the forwarder
11.     End if
12. End if
13. Calculate reputation-based on C2C for both sender and forwarder
```

Using this framework where the sender, forwarder and neighbor nodes take part in the reputation calculation, the false accusation problem can be avoided. In false accusation, nodes spread false reputation values about other nodes. Our scheme can efficiently handle this problem. For example, in Figure 1, when a sender node *S* transmits a packet to a

forwarder node F, both the sender S and the neighbor node M update the contribution of the F as the forwarder node F forwards the message. As a result, both the sender and the neighbor nodes have the same value of contribution for node F. Similarly, for sender S, the F and M have the same consumption value.

3.2.3. The Reputation Module

After receiving the C2C information from the monitor, the reputation system calculates the value of $R_1^n$and R_2^n for each node n and compares it with the reputation threshold. If the reputation value is less than the threshold, the node is then added to the selfish list which contains all selfish nodes. The reputation module finally passes the selfish list to the path manager.

3.2.4. The Path Manager

This unit is responsible for finding a route without passing through a selfish node(s) to the destination. After receiving the selfish list from the reputation module, it checks if the selfish node exists along the path, then the path manager avoids the path over that node(s).

3.2.5. Selfish Node Isolation Unit

Each node within the network will check the address of the source node of the packets into its selfish list; if the address of the sender node exists in the selfish list, then the packets from that node will be discarded, otherwise the packets will be forwarded to the next node.

3.3. Computational Complexity of the Proposed Scheme

The proposed reputation-based scheme has a computational complexity of $O(N \times M)$, where N is the number of nodes in the network and M is the number of packets to be checked (consumption or contribution) for each node. Therefore, the computational complexity is linear with N. It is worth noting that the proposed scheme is fully distributed, and the nodes do not share C2C information with neighbors; hence, there is no communication overhead.

4. Results and Discussion

In this section, the proposed reputation-based scheme is evaluated through simulation to determine its efficiency. The results are presented in the next section.

4.1. Simulation Setup

The proposed scheme is simulated using network simulator NS-2 and compared with OCEAN to check the efficiency of our scheme using the simulation parameters given in Table 2. The AWK scripts are developed for the collection of data from NS-2 simulator trace files. To automate the process to run extensive scenarios without user interference, we developed a bash script. The random waypoint model is used as a mobility model and CBR is used for UDP traffic generation. The simulation is conducted for different area sizes and pause times. The simulation time, speed of nodes, and number of selfish nodes are varied in order to perform an in-depth evaluation of the scheme. The simulation results are averaged over 30 randomly generated scenarios.

4.2. Performance Metrics

To evaluate and compare the proposed scheme with OCEAN, the following performance metrics are used.

- Good throughput—indicates the throughput available to good or regular nodes. It determines the ratio between the numbers of the packets received at the destination to the number of packets sent by the sender node, as denoted in Equation (4):

$$Good\ Throughput = \frac{\sum (Received\ pkts)}{\sum (Sent\ pkts\ by\ good\ nodes)} \quad (4)$$

Table 2. Different simulation parameters.

Parameter	Value
Area	500, 1000, 1500, 2000 m^2
Pause Time	10, 20, 30, 40, 50, 60 s
Maximum Speed	5, 10, 15, 20, 25, 30 m/s
Number of Nodes	50
Number of Connections	30
Application	CBR
Packet Size	64 B
Simulation Time	100, 200, 300, 400, 500, 600 s
Mobility Model	Random Waypoint Model
Selfish Population	0% to 100%

- Evil throughput—determines the throughput offered to misbehaving nodes:

$$Evil\ Throughput = \frac{\sum (Received\ pkts)}{\sum (Sent\ pkts\ by\ misbehaving)} \quad (5)$$

- Packet delivery fraction—the percentage between the number of packets originated by CBR sources and the number of packets received by CBR sinks at the destination;
- NRL—normalized routing load (NRL) is the fraction of packets communicated at the network layer to the CBR packets at the application layer by the receiver node;
- End-to-end delay—represents retransmission delay, queuing delay, buffering during route discovery, and propagation delay.

4.3. Attacker Types

Two types of attackers are created during our simulation-based experimentations, i.e., type-I and type-II attackers. Type-I attackers are those malicious nodes that take part in the route discovery process; once included in the routes, they start to act selfishly and drop packets. Type-II attackers, on the other hand, are malicious nodes that do not take part in the route discovery process at all, i.e., they drop the RREQ (route request) control packets; hence, they never act as intermediate nodes. These attackers are more crucial to be detected. In the following sections, the proposed scheme evaluation is presented where it is compared with OCEAN.

4.4. Simulation Results

4.4.1. Effect of Area

The impact of different terrain sizes on the performance of our proposed scheme is evaluated in the presence of 20% of type-I and type-II attackers with mobility of 10 m/s and a pause time of 60 s.

- Performance in the presence of type-I attackers: The performance in terms of packet drops is given in Figure 2a. It can be seen that OCEAN reported small packet drops against small areas, however, as the area increases our scheme outperforms OCEAN because the benchmark scheme provides a second chance to selfish nodes to join the network. The end-to-end delay of both schemes is shown in Figure 2b. OCEAN reported less delay as compared to our scheme for all terrain sizes because OCEAN uses a rank-based routing mechanism. Figure 2c shows the evil throughput, in which our scheme outperforms the benchmark scheme and recoded low evil throughput. Figure 2d represents good throughput, almost a similar performance can be seen in terms of good throughput. NRL is depicted in Figure 2e; NRL of our scheme is less as compared to OCEAN. Figure 2f represents packet delivery fraction; our scheme produces a higher PDF than OCEAN.

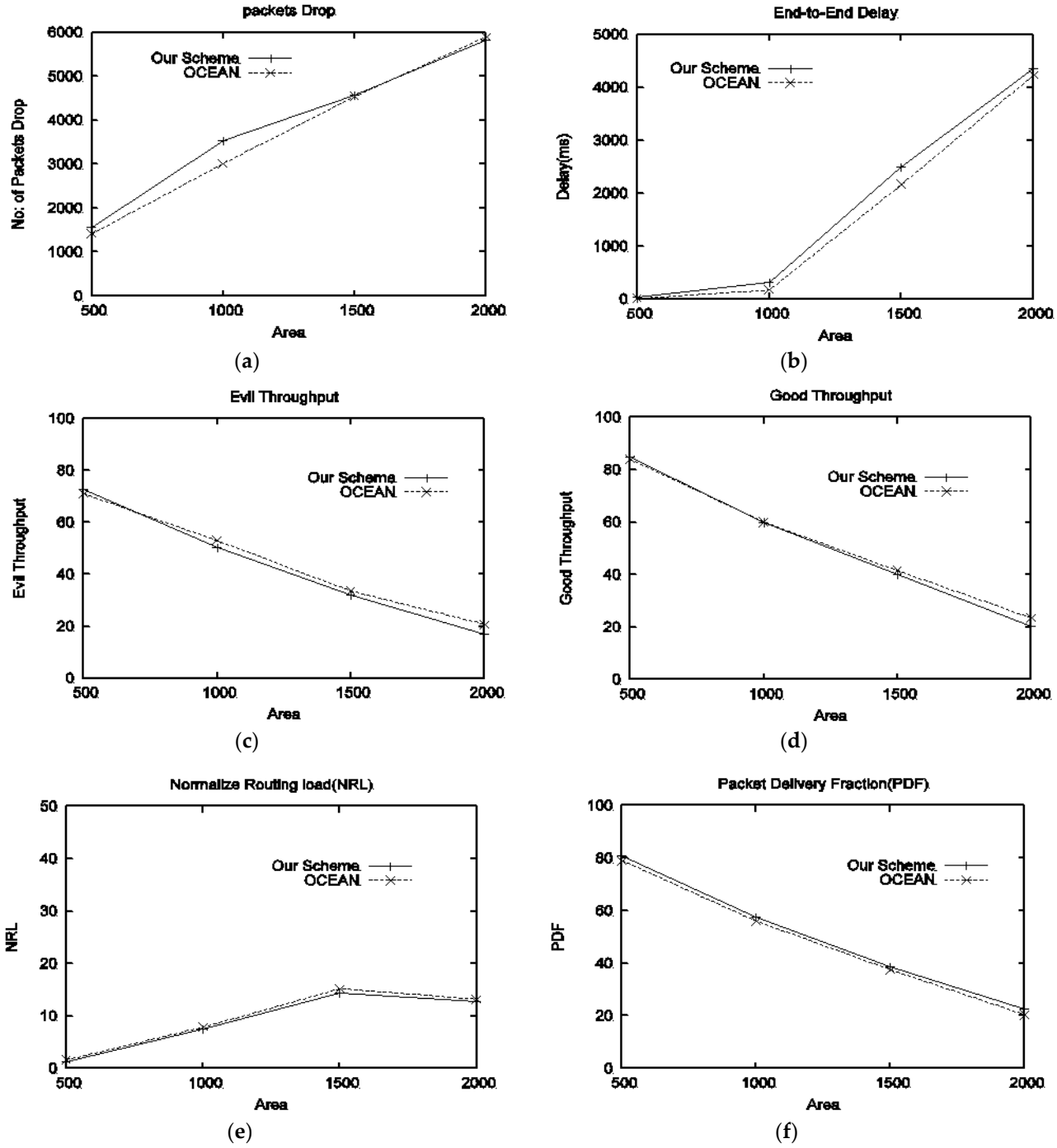

Figure 2. Performance of the proposed scheme with respect to different sizes of the area against type-I attackers: (**a**) drop packets vs. area, (**b**) delay vs. area, (**c**) evil throughput vs. area, (**d**) good throughput vs. area, (**e**) NRL vs. area, (**f**) PDF vs. area.

- Performance in the presence of type-II attackers: Figure 3a shows the end-to-end delay, in the small area both schemes produce the same delay, but with an increase in area. OCEAN's delay is going to decrease, because of previously stored routing information. Packet drop is shown in Figure 3b; it can be witnessed that OCEAN has a small packet drop because our scheme suffers from a hard/bad location problem, i.e., our scheme uses consumption to contribution ratio for reputation calculation, thus if a node is located at the boundary, then every time that node will appear as a source node and consequently that node consumption will be increasing, which will decrease a node's

reputation. Figure 3c represents evil throughput. With all area sizes, evil throughput is zero for both schemes because, during the route establishment phase, type-II attackers do not participate in RREQ. Good throughput is shown in Figure 3d; according to the results with large terrain size our scheme performs well in terms of good throughput as compared to OCEAN. Normalized routing load is presented in Figure 3e; it can be seen that up to 1000 m^2 areas produce the same results; however, with large size area, OCEAN offered high NRL. PDF is depicted in Figure 3f; our scheme delivered more packets as compared to OCEAN.

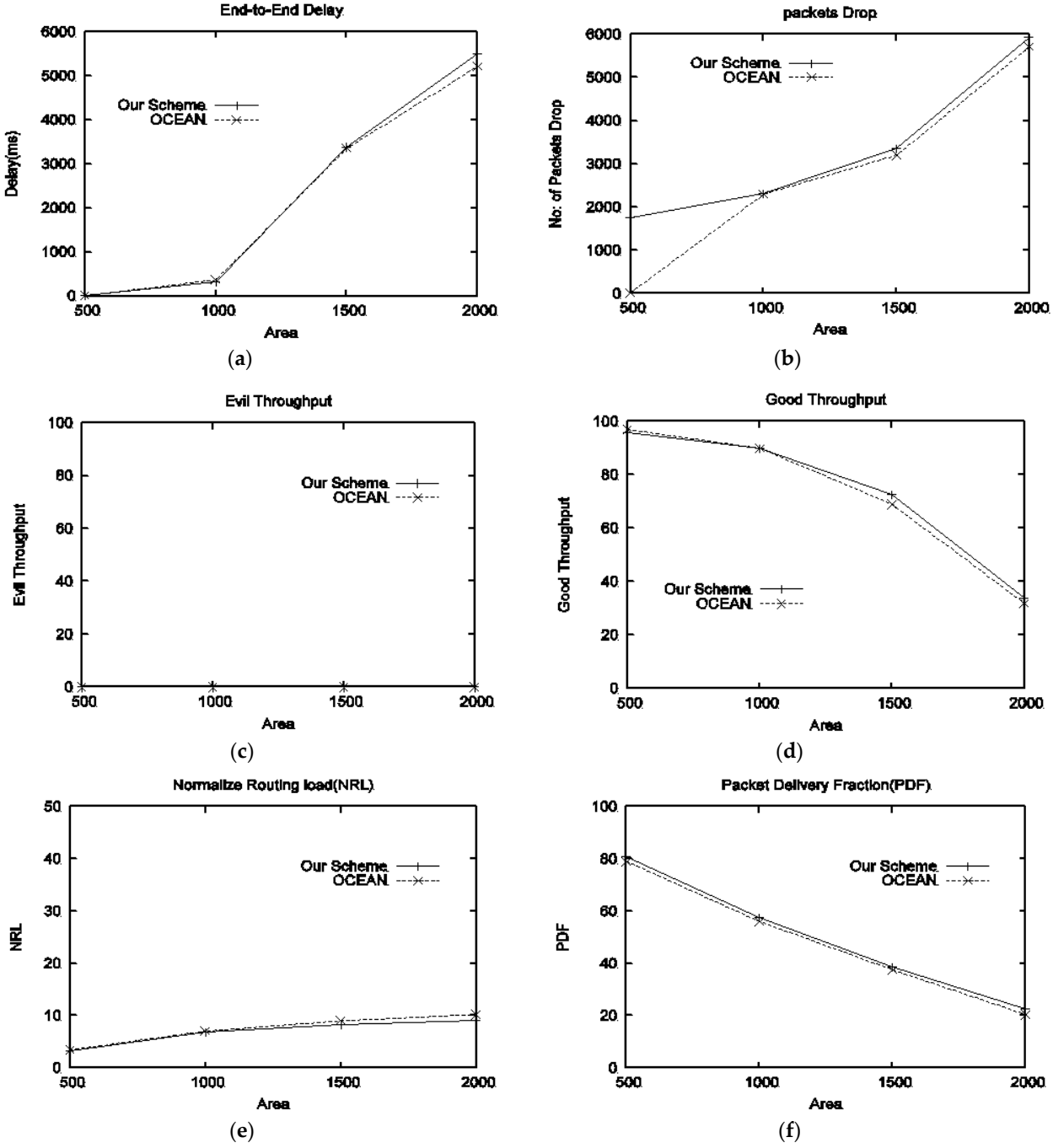

Figure 3. Performance of proposed scheme with respect to different sizes of the area against type-II attackers: (**a**) delay vs. area, (**b**) drop packets vs. area, (**c**) evil throughput vs. area, (**d**) good throughput vs. area, (**e**) NRL vs. area, (**f**) PDF vs. area.

4.4.2. Effect of Mobility

The effect of speed or node mobility on the performance of both schemes within a terrain size of 100 m^2 is explained as follows.

- Performance in the presence of type-I attackers: Figure 4a represents end-to-end delay; it can be seen that our scheme produces greater delay than OCEAN because OCEAN uses former information for route selection, i.e., OCEAN avoids paths through misbehavior nodes; however, our scheme assesses new neighbors from the start. This problem will be further investigated in our future work. The packet drop of both schemes is shown in Figure 4b; it can be observed that OCEAN drops fewer packets with low speed, but as mobility increases, OCEAN drops more packets as compared to our scheme. The reason behind the scene is the second-chance mechanism of OCEAN, as a malicious node joins the network, it again initiates packet dropping attack. Evil throughput is presented in Figure 4c; it can be seen that with low mobility, evil throughput of OCEAN is small, but with medium speed, both schemes are the same; however, with high-speed, evil throughput of our scheme decreases. Good throughput is shown in Figure 4d, which shows that our scheme provides high throughput to good nodes. Normalized routing load is shown in Figure 4e; it can be concluded that NRL of our scheme is less than OCEAN from low to high mobility. PDF is shown in Figure 4f; our scheme delivered more packets than OCEAN.
- Performance in the presence of type-II attackers: Effect of mobility in the presence of type-II attackers against different mobility on the performance of both schemes in terms of delay, packet drop, evil throughput, good throughput, normalized routing load and packet delivery fraction is presented in Figure 5a–f. Delay is given in Figure 5a; our scheme results in higher delay as compared to OCEAN. Figure 5b represents drop packets; it can be found that our scheme leads to a slightly greater packet drop because of a bad/hard location problem. The throughput available to evil nodes is presented in Figure 5c; a zero evil throughput of both schemes can be observed, because selfish attackers discard RREQ packets. Good throughput is shown in Figure 5d; our scheme provides more throughput to good nodes as compared to OCEAN. Figure 5e represents NRL; it can be observed that our scheme's NRL is small compared to OCEAN. PDF is depicted in Figure 5f; our scheme performs well compared to OCEAN.

4.4.3. Effect of Malicious Attack

Both schemes are investigated under various percentages of malicious nodes. The behavior of both schemes in terms of delay, packet drop, evil throughput, good throughput, NRL, and PDF is shown in Figure 6a–f. End-to-end delay is presented in Figure 6a; small delay of our scheme can be observed in the presence of 50% malicious nodes, however, with an increase in malicious percentage same behavior of both schemes can be observed. Packet drop is presented in Figure 6b; OCEAN leads to high packet drop in comparison with our scheme, as malicious node(s) once again join the network using the second-chance mechanism. Figure 6c shows the throughput available to evil nodes; however, our scheme performs well as the malicious percentage increases. The throughput provided to good nodes by both schemes is given in Figure 6d; our scheme is better than OCEAN, particularly with a 60% malicious percentage. NRL is presented in Figure 6e; as the malicious percentage is above 50%, our scheme NRL decreases; however, the same behavior of both schemes can be observed when the malicious percentage is below 50%. Packet delivery fraction is given in Figure 6f; both schemes delivered the same overall malicious packets percentage.

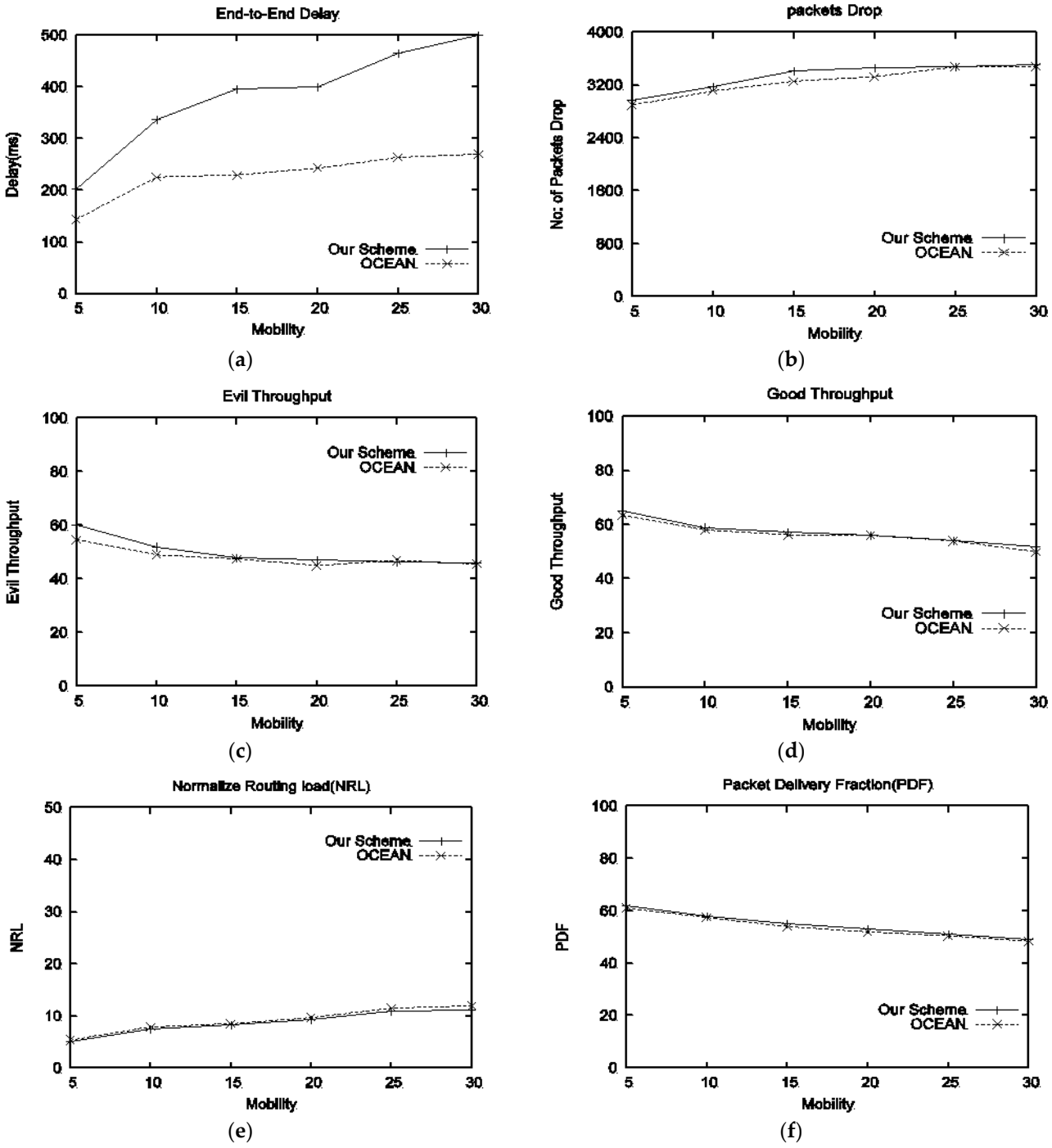

Figure 4. Performance comparison in the presence of type-1 attackers with respect to different speeds of the nodes: (**a**) delay vs. mobility, (**b**) average drop packets vs. mobility, (**c**) evil throughput vs. mobility, (**d**) good throughput vs. mobility, (**e**) NRL vs. mobility, (**f**) PDF vs. mobility.

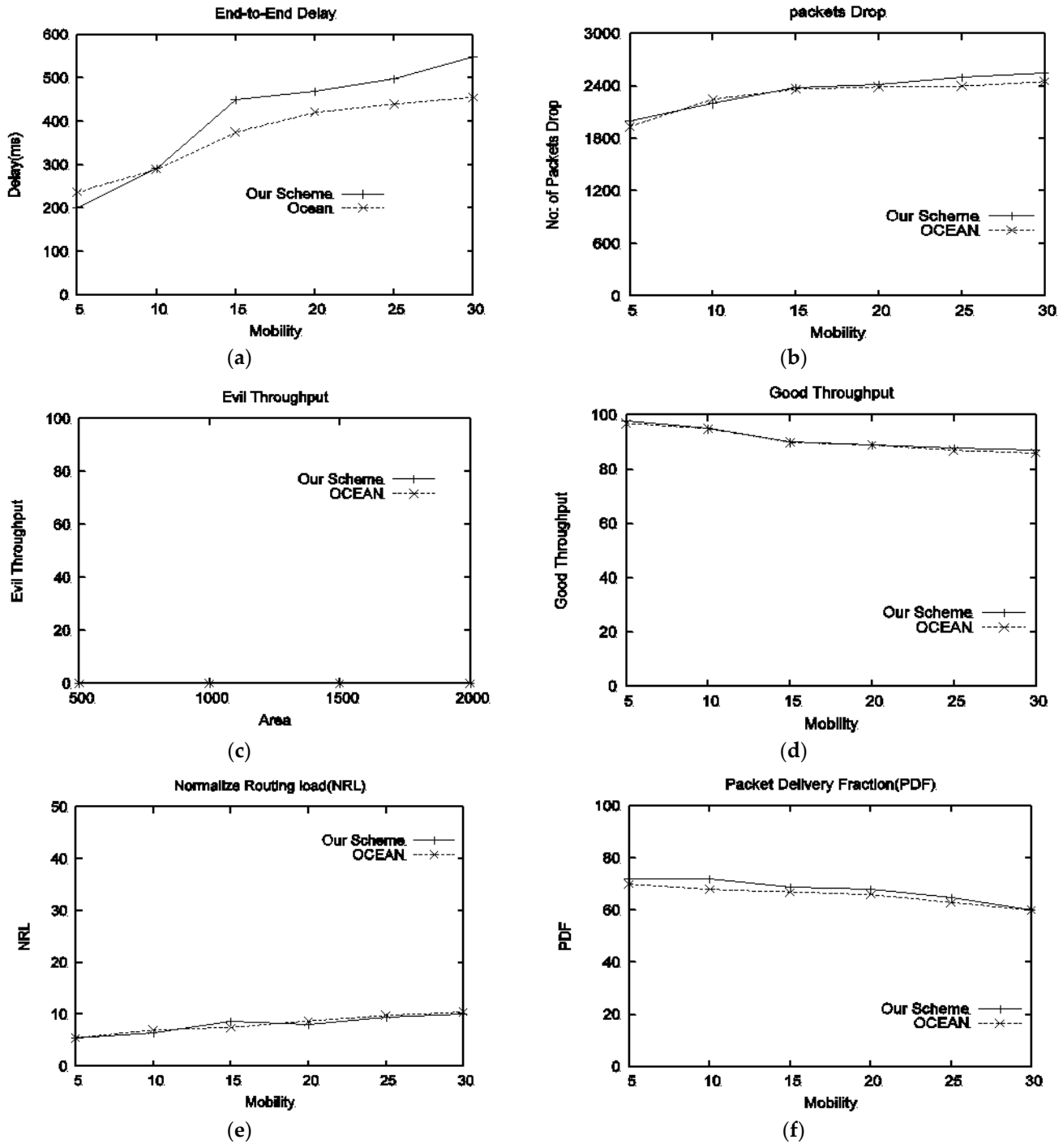

Figure 5. Performance in the presence of type-II attackers with respect to different speeds: (**a**) delay vs. mobility, (**b**) average drop packets vs. mobility, (**c**) evil throughput vs. mobility, (**d**) good throughput vs. mobility, (**e**) NRL vs. mobility, (**f**) PDF vs. mobility.

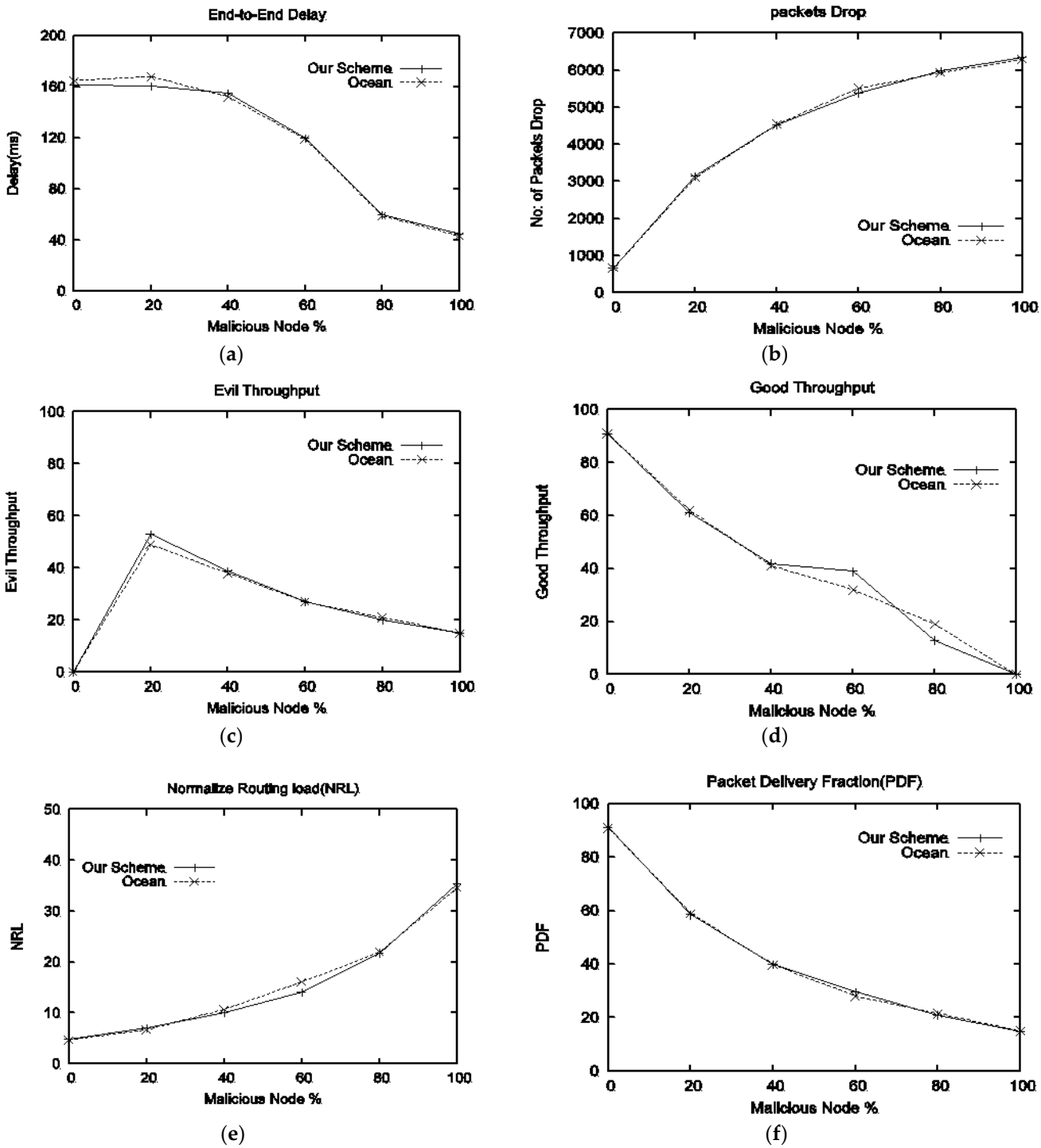

Figure 6. Performance comparison with respect to varying number of malicious percentage: (**a**) delay vs. malicious nodes, (**b**) average drop packets vs. malicious modes, (**c**) evil throughput vs. malicious nodes, (**d**) good throughput vs. malicious nodes, (**e**) NRL vs. malicious nodes, (**f**) PDF vs. malicious nodes.

4.4.4. Effect of Selfish Attack

Results of both schemes in terms of delay, packet drop, evil throughput, good throughput, NRL, and PDF against the different percentages of selfish attackers are shown in Figure 7a–f. End-to-end delay is presented in Figure 7a; our scheme produces a minimal delay as compared to OCEAN. Packet drop is given in Figure 7b; a small amount of packet

drop can be observed in our scheme throughout all selfish percentages, as our scheme does not allow a selfish node again once detected, while OCEAN gives a second chance to selfish nodes to participate in the network. Figure 7c represents evil throughput; both schemes show a zero evil throughput, as nodes do not take part in the REEQ and RREP phase. Figure 7d represents good throughput; it can be observed that our scheme performs better than OCEAN in the presence of selfish percentage ranging from 0% to 100%. NRL is presented in Figure 7e; NRL produced by our scheme is less than OCEAN along with all selfish percentages. PDF is shown in Figure 7f; it can be witnessed that our scheme outperforms OCEAN in terms of PDF.

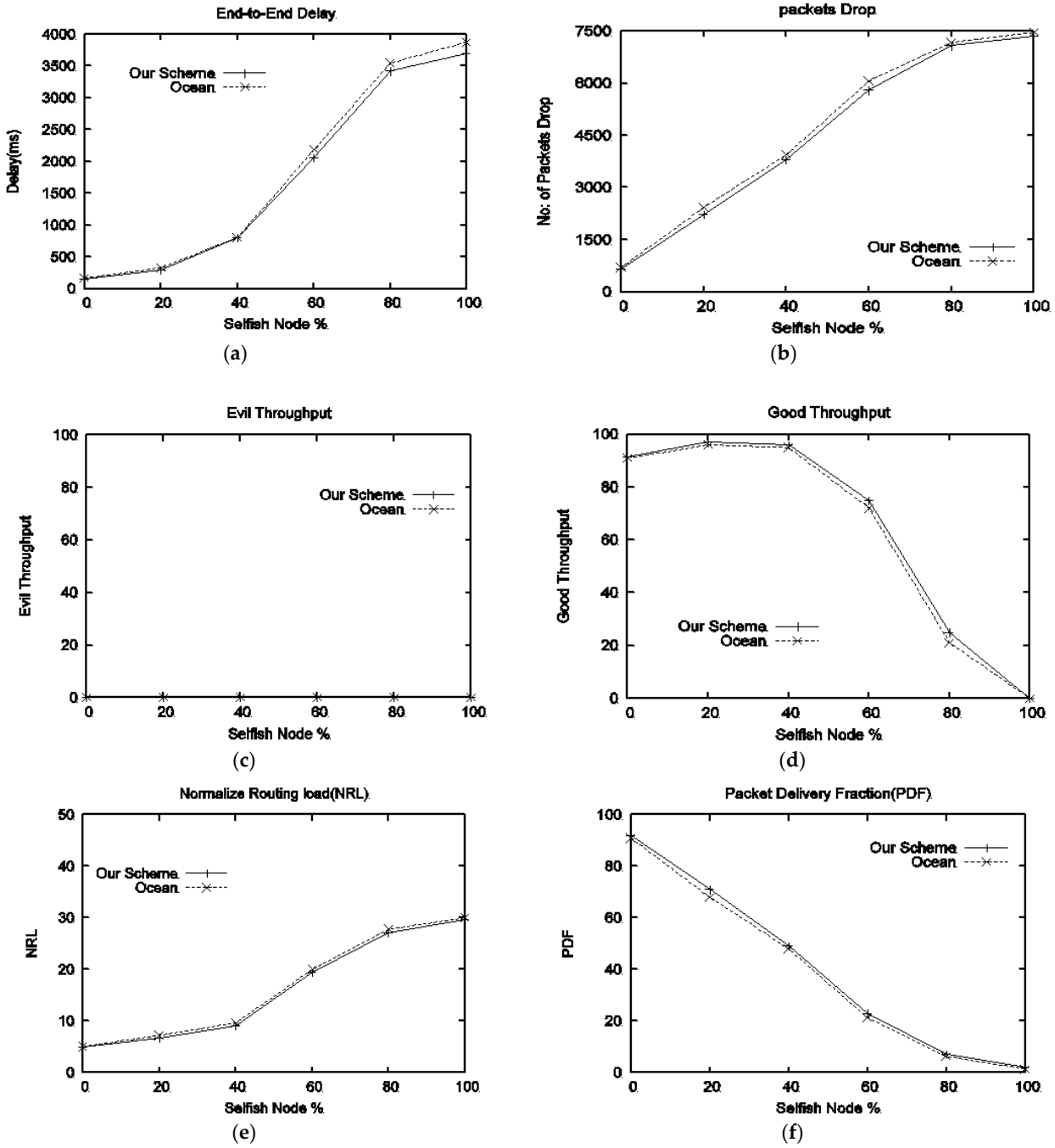

Figure 7. Performance comparison with respect to varying selfish nodes percentage: (**a**) delay vs. selfish nodes, (**b**) average drop packets vs. selfish nodes, (**c**) evil throughput vs. selfish nodes, (**d**) good throughput vs. selfish nodes, (**e**) NRL vs. selfish nodes, (**f**) PDF vs. selfish nodes.

4.4.5. Detection Accuracy (Type-I Attack)

The detection accuracy of our scheme and OCEAN for different simulation times is shown in Figure 8. As compared to OCEAN, our scheme detects all malicious nodes within less time because OCEAN cannot detect such nodes that partially drop some packets to sustain its reputation threshold. In detection accuracy, our scheme outperforms OCEAN and isolates malicious nodes from the network within the shortest time.

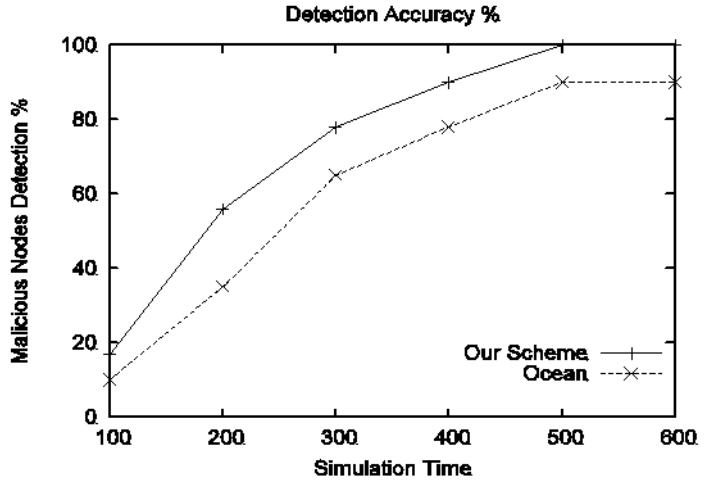

Figure 8. Performance comparison of the proposed scheme and OCEAN in terms of detection accuracy in the presence of type-I attackers.

4.4.6. Detection Accuracy (Type-II Attack)

Results in terms of detection accuracy in the presence of type-II attackers of both schemes are given in Figure 9. It can be seen that OCEAN detection accuracy is more stable than our scheme, because our scheme uses the idea of consumption to contribution ratio that suffers from the hard/bad location problem. For example, if a node is located at a position or area where there is no chance or less chance to forward packets for other nodes, then every time that node will appear as a source node. This position will increase node-sending packets (consumption) and eventually that node reputation will be condensed, and as the reputation falls below the given predefined threshold, it will be considered as a selfish node. Hence, OCEAN outperforms our scheme in terms of detection accuracy in the presence of selfish attackers.

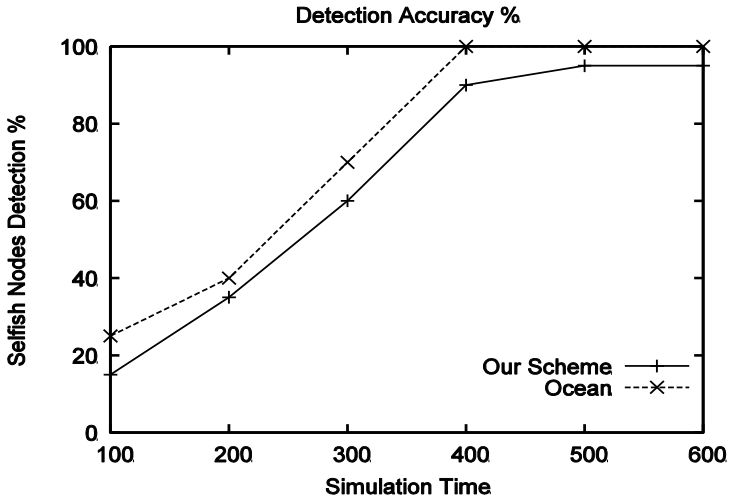

Figure 9. Performance comparison of the proposed scheme and OCEAN in terms of detection accuracy in the presence of type-II attackers.

4.4.7. Scalability of the Proposed Scheme (Type-II Attack)

To assess the scalability of the proposed scheme, different numbers of nodes are used. The selfish population is fixed at 20% and the speed is kept at 15 m/s. Figure 10 shows the end-to-end delay versus the number of nodes. It is evident that our proposed algorithm performs similarly regardless of the number of nodes and reports less delay as compared to the benchmark scheme. It is worth noting that the selfish nodes continuously change their position (location), but our scheme can effectively locate these nodes and isolate them from the network quickly. As a result, the nodes can quickly reconstruct the route and communicate within a controlled delay. In contrast, when nodes in OCEAN change location, every time they appear as new nodes they will utilize network resources until they are detected. As a result, it can be concluded that the proposed scheme can scale to networks with a large number of users.

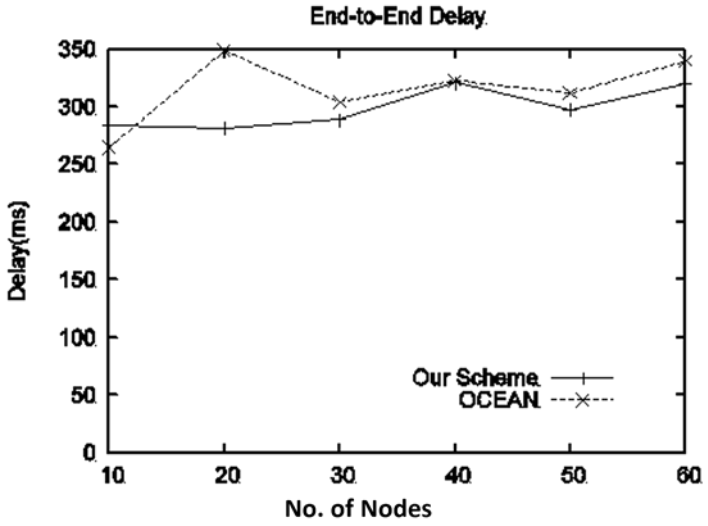

Figure 10. Scalability of the proposed scheme with respect to fixed speed, fixed selfish population, and varying number of mobile nodes.

5. Conclusions and Future Work

In this research, a novel and simple reputation-based scheme was proposed that utilized the consumption-to-contribution ratio of each node in the network to detect and isolate the selfish nodes in MANETs. The proposed scheme was compared with the benchmark scheme, namely, OCEAN. The simulation results showed that the proposed reputation-based scheme outperformed the benchmark scheme in terms of packet delivery fraction, normalized routing load, and good throughput in the presence of malicious and selfish nodes. However, OCEAN performed well in terms of delay as compared to the proposed scheme. Furthermore, it was found that the proposed algorithm enhanced the detection accuracy of selfish nodes as compared to OCEAN. The proposed scheme suffered from a bad or hard location problem, which degraded the detection accuracy in the presence of selfish nodes.

Future research directions include comparing the proposed scheme with some of the most recent reputation-based schemes, and incorporating machine learning, including deep reinforcement learning, to solve hard or bad location problems and enhance detection accuracy. In addition, investigating the congestion effects on the proposed scheme and reputation rebuilding mechanisms will be explored, which can lead to a node that was perceived as selfish and isolated from the network being reevaluated and reintegrated into it.

Author Contributions: Conceptualization, M.F. (Muhammad Fayaz, m.fayaz@qmul.ac.uk), J.G., and S.A.; methodology, M.F. (Muhammad Fayaz, muhammad.fayaz@ucentralasia.org), G.M. and J.G.; software, M.F. (Muhammad Fayaz, m.fayaz@qmul.ac.uk), J.G., and G.M.; validation, A.K., and S.A.; formal analysis, M.F. (Muhammad Fayaz, m.fayaz@qmul.ac.uk) and J.G.; investigation, M.F. (Muhammad Fayaz, muhammad.fayaz@ucentralasia.org); resources, G.M.; data curation, M.F.; writing—original draft preparation, M.F. (Muhammad Fayaz, m.fayaz@qmul.ac.uk) and J.G.; writing—review and editing, S.A. and J.G.; visualization, A.K.; supervi-sion, A.K.; administration, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the "Regional Innovation Strategy (RIS)" through the NRF and funded by the Ministry of Education (MOE) (2021RIS-001).

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

References

1. Fu, X.; Fortino, G.; Pace, P.; Aloi, G.; Li, W. Environment-fusion multipath routing protocol for wireless sensor networks. *Inf. Fusion* **2020**, *53*, 4–19. [CrossRef]
2. Fu, X.; Yang, Y. Modeling and analyzing cascading failures for Internet of Things. *Inf. Sci.* **2021**, *545*, 753–770. [CrossRef]
3. Fayaz, M.; Rahman, Z.U.; Rahman, M.U.; Abbas, S. Effect Of Terrain Size And Pause Time On The Performance Of Reactive Routing Protocols. *J. Teknol.* **2016**, *78*. [CrossRef]
4. Kaur, G.; Thakur, P. Routing Protocols in MANET: An Overview. In Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 5–6 July 2019; Volume 1, pp. 935–941.
5. Sakiz, F.; Sen, S. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. *Ad Hoc Netw.* **2017**, *61*, 33–50. [CrossRef]
6. Conti, M.; Giordano, S. Mobile ad hoc networking: Milestones, challenges, and new research directions. *IEEE Commun. Mag.* **2014**, *52*, 85–96. [CrossRef]
7. Mehmood, G.; Khan, M.Z.; Abbas, S.; Faisal, M.; Rahman, H.U. An Energy-Efficient and Cooperative Fault- Tolerant Communication Approach for Wireless Body Area Network. *IEEE Access* **2020**, *8*, 69134–69147. [CrossRef]
8. Wang, Y.; Fu, X.; Yang, Y.; Postolache, O. Analysis on Cascading Robustness of Energy-balanced Scale-free Wireless Sensor Networks. *AEU-Int. J. Electron. Commun.* **2021**, *140*, 153933. [CrossRef]
9. Sangoleye, F.; Irtija, N.; Tsiropoulou, E.E. Data Acquisition in Social Internet of Things based on Contract Theory. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
10. Jim, L.E.; Islam, N.; Gregory, M.A. Enhanced MANET security using artificial immune system based danger theory to detect selfish nodes. *Comput. Secur.* **2021**, *113*, 102538. [CrossRef]
11. Tripathy, B.K.; Jena, S.K.; Reddy, V.; Das, S.; Panda, S.K. A novel communication framework between MANET and WSN in IoT based smart environment. *Int. J. Inf. Technol.* **2021**, *13*, 921–931. [CrossRef]
12. Bounouni, M.; Bouallouche-Medjkoune, L. Acknowledgment-based punishment and stimulation scheme for mobile ad hoc network. *J. Supercomput.* **2018**, *74*, 5373–5398. [CrossRef]
13. Zhang, W.; Zhu, S.; Tang, J.; Xiong, N. A novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection in wireless sensor networks. *J. Supercomput.* **2017**, *74*, 1779–1801. [CrossRef]
14. Sangi, A.R.; Liu, J.; Alkatheiri, M.S.; Anamalamudi, S. Secure opinion sharing for reputation-based systems in mobile ad hoc networks. *Meas. Control* **2020**, *53*, 748–756. [CrossRef]
15. Nobahary, S.; Garakani, H.G.; Khademzadeh, A.; Rahmani, A.M. Selfish node detection based on hierarchical game theory in IoT. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 255. [CrossRef]
16. Osseiran, O.E.A.; Song, J.; Monserrat, J.F. Internet of Things. *IEEE Commun. Stand. Mag.* **2017**, *1*, 84. [CrossRef]
17. Abawajy, J.H.; Hassan, M.M. Federated Internet of Things and Cloud Computing Pervasive Patient Health Monitoring System. *IEEE Commun. Mag.* **2017**, *55*, 48–53. [CrossRef]
18. Abbas, S. *A Layered Security Approach for Cooperation Enforcement in MANETs*; Liverpool John Moores University: Liverpool, UK, 2011.
19. Buttyán, L.; Hubaux, J.-P. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *Mob. Netw. Appl.* **2003**, *8*, 579–592. [CrossRef]
20. Abbas, S.; Merabti, M.; Llewellyn-Jones, D. On the evaluation of reputation and trust-based schemes in mobile ad hoc networks. *Secur. Commun. Netw.* **2015**, *8*, 4041–4052. [CrossRef]
21. Thorat, S.A.; Kulkarni, P.J. Opportunistic Routing in Presence of Selfish Nodes for MANET. *Wirel. Pers. Commun.* **2015**, *82*, 689–708. [CrossRef]
22. Figueiredo, D.R.; Garetto, M.; Towsley, D. Exploiting mobility in ad-hoc wireless networks with incentives. *Relatório Técnico* **2004**, 4–66. Available online: https://web.cs.umass.edu/publication/docs/2004/UM-CS-2004-066.pdf (accessed on 29 October 2021).
23. Marti, S.; Giuli, T.J.; Lai, K.; Baker, M. Mitigating routing misbehavior in mobile ad hoc networks. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking-MobiCom '00, ACM, Boston, MA, USA, 6–11 August 2000; pp. 255–265.

24. Buchegger, S.; le Boudec, J.-Y. Performance analysis of the CONFIDANT protocol. In Proceedings of the 3rd ACM Interna-tional Symposium on Mobile Ad Hoc Networking & Computing, Lausanne, Switzerland, 9–11 June 2002; pp. 226–236.
25. Michiardi, P.; Molva, R. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *Advanced Communications and Multimedia Security*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 107–121.
26. Hu, J.; Burmester, M. LARS: A locally aware reputation system for mobile ad hoc networks. In Proceedings of the 44th Annual Southeast Regional Conference, Melbourne, FL, USA, 10–12 March 2006; pp. 119–123.
27. Bansal, S.; Baker, M. Observation-based cooperation enforcement in ad hoc networks. *arXiv* **2003**, arXiv:cs/0307012.
28. Buttyan, L.; Hubaux, J.-P. *Nuglets: A Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks*; Swiss Federal Institute of Technology: Lausanne, Switzerland, 2001.
29. Zhong, S.; Chen, J.; Yang, Y. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In Proceedings of the IEEE INFOCOM Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), San Francisco, CA, USA, 30 March–3 April 2003; pp. 1987–1997. [CrossRef]
30. Djahel, S.; Nait-Abdesselam, F.; Zhang, Z. Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges. *IEEE Commun. Surv. Tutor.* **2010**, *13*, 658–672. [CrossRef]
31. Abbas, S.; Merabti, M.; Llewellyn-Jones, D. The effect of direct interactions on reputation based schemes in mobile ad hoc networks. In Proceedings of the 2011 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2011; pp. 297–302.
32. Buchegger, S.; le Boudec, J. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. In Proceedings of the WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia-Antipolis, France, 3–5 March 2003.
33. Jesudoss, A.; Raja, S.K.; Sulaiman, A. Stimulating truth-telling and cooperation among nodes in VANETs through payment and punishment scheme. *Ad Hoc Netw.* **2015**, *24*, 250–263. [CrossRef]
34. Mani, P.; Kamalakkannan, P. Mitigating selfish behavior in mobile ad hoc networks: A survey. *Int. J. Comput. Appl.* **2013**, *73*, 1–7.
35. Jakobsson, M.; Hubaux, J.-P.; Buttyán, L. A Micro-Payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 15–33.
36. Ben Salem, N.; Buttyan, L.; Hubaux, J.-P.; Jakobsson, M. Node cooperation in hybrid ad hoc networks. *IEEE Trans. Mob. Comput.* **2006**, *5*, 365–376. [CrossRef]
37. Mahmoud, M.E.; Shen, X. An Integrated Stimulation and Punishment Mechanism for Thwarting Packet Dropping Attack in Multihop Wireless Networks. *IEEE Trans. Veh. Technol.* **2011**, *60*, 3947–3962. [CrossRef]
38. Mahmoud, M.M.E.A.; Shen, X. FESCIM: Fair, Efficient, and Secure Cooperation Incentive Mechanism for Multihop Cellular Networks. *IEEE Trans. Mob. Comput.* **2011**, *11*, 753–766. [CrossRef]
39. Samian, N.; Zukarnain, Z.A.; Seah, W.K.; Abdullah, A.; Hanapi, Z.M. Cooperation stimulation mechanisms for wireless multihop networks: A survey. *J. Netw. Comput. Appl.* **2015**, *54*, 88–106. [CrossRef]
40. Abdel-Fattah, F.; Farhan, K.A.; Al-Tarawneh, F.H.; AlTamimi, F. Security Challenges and Attacks in Dynamic Mobile Ad Hoc Networks MANETs. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 28–33.
41. Abbas, S.; Merabti, M.; Llewellyn-Jones, D. Deterring whitewashing attacks in reputation based schemes for mobile ad hoc networks. In Proceedings of the 2010 IFIP Wireless Days, Venice, Italy, 20–22 October 2010; pp. 1–6.
42. Ourouss, K.; Naja, N.; Jamali, A. Defending Against Smart Grayhole Attack Within MANETs: A Reputation-Based Ant Colony Optimization Approach for Secure Route Discovery in DSR Protocol. *Wirel. Pers. Commun.* **2021**, *116*, 207–226. [CrossRef]
43. M. Ponnusamy, Detection of Selfish Nodes Through Reputation Model In Mobile Adhoc Network-MANET. *Turk. J. Comput. Math. Educ.* **2021**, *12*, 2404–2410.
44. Mehmood, G.; Khan, M.S.; Waheed, A.; Zareei, M.; Fayaz, M.; Sadad, T.; Kama, N.; Azmi, A. An Efficient and Secure Session Key Management Scheme in Wireless Sensor Network. *Complex* **2021**, *2021*, 6577492. [CrossRef]