Research Article

An SETM Algorithm for Combating SSDF Attack in Cognitive Radio Networks

S. Tephillah and J. Martin Leo Manickam

1Department of Electronics and Communication, St. Joseph’s Institute of Technology, St. Joseph’s Group of Institutions, Anna University, Chennai, Tamil Nadu, India
2Department of Electronics and Communication, St. Joseph’s College of Engineering, St. Joseph’s Group of Institutions, Anna University, Chennai, Tamil Nadu, India

Correspondence should be addressed to S. Tephillah; tephisam@gmail.com

Received 23 January 2020; Revised 13 April 2020; Accepted 4 June 2020; Published 21 July 2020

Abstract

Security is a pending challenge in cooperative spectrum sensing (CSS) as it employs a common channel and a controller. Spectrum sensing data falsification (SSDF) attacks are challenging as different types of attackers use them. To address this issue, the sifting and evaluation trust management algorithm (SETM) is proposed. The necessity of computing the trust for all the secondary users (SUs) is eliminated based on the use of the first phase of the algorithm. The second phase is executed to differentiate the random attacker and the genuine SUs. This reduces the computation and overhead costs. Simulations and complexity analyses have been performed to prove the efficiency and appropriateness of the proposed algorithm for combating SSDF attacks.

1. Introduction

Exploitation of a licensed spectrum by unlicensed users without interference is a prodigy for wireless communications. Cognitive radio technology (CR) has introduced the issue of the utilisation of the spectrum based on the process of dynamic spectrum access (DSA). The CR is anticipated to be a vital technology for fifth-generation (5G) communications. The CR technology enables the unlicensed or secondary users (SUs) to access opportunistically the prescribed channels of the licensed or primary users (PUs) without interference. DSA involves spectral sensing, analysis, and allocation by sensing the white spaces (holes) in the frequency bands. Cooperative spectrum sensing (CSS) increases the accuracy of the PU occupancy detection by collecting sensing information from the SUs [1]. However, CSS paves the way for malicious attacks that modify the sensing reports given that the channel used for reporting is common.

The alteration of the sensing report by malicious users (MU) is the spectrum data falsification attack (SSDF) that typically occurs at the data link layer. The SSDF attack can be classified into five classes [2]: (a) always yes: in the absence or presence of the PU, the malicious user always reports the presence of a PU to divest the genuine SU; (b) always no: to induce interference, the malicious user always reports the absence of the PU even in its presence; (c) alternator: the malicious user always conflicts the genuine SUs detection output; (d) selfish: at times, the malicious attacker does not participate in the sensing process to save energy; and (e) contingent: report data falsely in a random manner. They play smartly such that identification becomes difficult.

2. Methodology

Trust is a firm belief in reliability. Thus, the genuine nature of the user (SU) can be quantified by measuring the degree of trust. However, the evaluation of trust for all the SUs may increase the complexity with respect to time and space. To address this particular issue in trust-based security systems, a new algorithm is proposed in this study referred to as sifting and evaluation trust management (SETM). The algorithm comprises two phases: (i) sifting and (ii) evaluation. In the first phase, identification of MUs subject to the first four categories is achieved by screening the sensing reports of the SUs with...
the fusion centre (FC) outputs over a period of time. Thus, most of the MUs are filtered in this phase. The second phase aids in the identification of contingent attackers from the genuine SUs by evaluating the trust of different categories, such as past events, registries, requites, and reliabilities. Additionally, the efficiency and appropriateness of SETM are proven using simulations and complexity analyses.

The rest of the study is organised as follows: Section 3 introduces related works pertaining to trust-based systems. Section 4 provides insights on the system model, the spectrum sensing process, and the identification of MUs. Section 5 provides the proposed SETM algorithm. Finally, the results, discussion, and conclusion sections are outlined in Sections 6 and 7.

3. Related Works

Chen et al. [3] have proposed a reputation-based cooperative spectrum sensing model in which the reputation is set by the FC for each SU based on the sensing results. The SU is then weighted according to the reputation value. However, the reputation degree is not adaptable. Kar et al. [4] proposed multi-weighted according to the reputation value. However, the rep-


cification of contingent attackers from the genuine SUs is identified based on the evaluation of history, action, incentive, and consistency trust. No profiteering technique is employed and selfish nodes cannot be identified. A trust-based on a

classification model. Section 4.1: Spectrum Sensing of SU. The presence and absence of SUs that participate in the sensing. But detection of all the suspicious level and consistency level of trust for the all SUs in a distributed manner. Feng et al. [5] have presented an XOR distance analysis to suppress the collusive SSDF attack. Zhao et al. have addressed a dynamic collusive SSDF attack based on a trust fluctuation clustering analysis to measure the similarities between two attackers that reflect the trust values [6]. A distance outlier detection approach is given by Singh et al. [7] for CSS–SSDF attacks. The methodology exploits the data stored in the FC to identify the MUs. The prior knowledge of the user is needed in the implementation of the method. Furthermore, an adaptive reputation evaluation for individual and collaborative SSDF attackers based on a linear weighted combination scheme has been proposed by Wang et al. [8]. Consistency measures are executed to detect outliers. Relying on a weighted scheme may not prove appropriate in random attack scenarios. Also, Wang et al. [9] have proposed a combatting technique for SSDF by calculating the suspicious level and consistency level of trust for the all the SUs that participate in the sensing. But detection of all types of SSDF attackers cannot be resolved by just measuring the consistency levels.

4. System Model

Let “K” be the number of SUs who sense the spectrum over a time slot “T,” and “M” be the number of MUs pertaining to all the attacking strategies. The system model is depicted in Figure 1. Additionally, the FC is considered to be the most powerful user with considerable potential and accumulated data to identify all types of SSDF attackers. The channel conditions are assumed to be perfect for transmission (control channels), and all the SUs use the energy detection technique to detect the presence and absence of the PU. Furthermore, the path loss is neglected given that the area of coverage of the CR system is assumed to be small [10].

4.1. Spectrum Sensing of SU. The presence and absence of PUs are determined by the FC based on hypothesis testing (statistical testing). Mathematically, it can be expressed according to Equations (1) and (2) as,

\[ H_0 : Y_s(k) = \omega_s(K) \quad (1) \]

\[ H_1 : Y_s(k) = \omega_s(K) + h_p \sigma P(K) \quad (2) \]

where \( Y_s(k) \) is the received signal, and \( \omega_s(K) \) is the white Gaussian noise function for “K” SUs \( (K = 1, 2, 3, \ldots, S) \). It is also considered that the noise has a zero mean and a variance \( \sigma \). \( P(K) \) is the \( K^{th} \) sample of the primary user signal detected by the SUs. \( H_0 \) represents the absenteeism of PU, and \( H_1 \) represents the existence of PU. Mathematically, Equation (3) represents the hypothesis.

\[ H_0 : \sum_{i=0}^{k} \delta_i \leq k \quad (3) \]

\[ H_1 : \sum_{i=0}^{k} \delta_i \geq k \]

In this study, a majority voting \( (K \text{ out of } N) \), hard decision fusion rule is applied. This rule has an advantage in terms of

![Figure 1: Representation of centralized cognitive radio system model.](image-url)
bandwidth and implementation. Each of the SUs sends a binary output to the FC indicating the presence (binary “1”) or absence (binary “0”) for the formulation of a final decision. FC decides in favour of the SUs if the desired output is confirmed by the SU. The FC calculates the received decision. FC decides in favour of the SUs if the SU’s decision is confirmed by the FC.

Mathematically expressed by Equation (5) [12].

The cumulative probability of detection and false alarm can be represented by Equation (7),

where \( y \) is the signal-to-noise ratio (SNR) of the received signal, and \( \lambda \) is the threshold, \( v \) is the product of time (\( T \)) and bandwidth (\( B \)), and \( Q_\nu \) represents the marqum Q function. The cumulative probability of detection and false alarm at the FC are represented by Equations (8) and (9).

\[
P_{d} = P_{r} \left\{ \sum_{i=1}^{s} \delta_{i} \geq K : H_{1} \right\}
\]

\[
P_{f} = P_{r} \left\{ \sum_{i=1}^{s} \delta_{i} < K : H_{0} \right\}
\]

4.2. Identification of MU’s. Trust is a time-varying parameter expressing the assured reliance on someone’s ability, strength, and truth. Thus, the study of past events of the SU would reveal their worthiness. In our proposed system, the trust analysis is pursued and completed in two phases, namely, the sifting and evaluation phases. In the sifting phase, the worthiness of the SU is observed based on the recording of past SU events over a period of time. Consider a sliding time window from 0 to \( T \) comprising equal time subplots denoted by \( t_{0}, t_{1}, \cdots, t_{k} \). During each timeslot, the spectrum sensing is completed by each SU. Let the sensing output of each SU be \( \{SU_{1}^{r}, SU_{2}^{r}, \cdots, SU_{s}^{r}\} \), and let the time of the observation range from \( t_{0} \) to \( t_{k} \), where \( k \in 2n + 1 \), where \( n \) is an integer. The FC accumulates all the reports in a record table, as shown in Table 1.

The FCO is the final decision by the FC based on the consideration of the K out of N rule. The details listed in different rows of the table show the output of “S” SUs at different timeslots, and the details listed in different columns show the sensing results of each SU.

5. The SETM Algorithm

5.1. Sifting Phase. In this study, the SETM algorithm is presented to identify all the aforementioned types of MUs, where \( S \) is the symbol used to denote sifting, and \( E \) is used to denote evaluation. As the name indicates, sifting means filtering. Therefore, by analysing the columns of the reports of each SU, it is possible to infer the nature of the malicious behaviours of the MUs. For example, by analysing the first column of the Table 1 for time slot \( t_{0} \) to \( t_{k} \), it can be inferred that the SU belongs to the MU class “always yes,” if all the columns have the sensing report set to binary “1.” After “\( K \)” timeslots, this phase can be continued to the next timeslot (\( t_{k} + 1 \) to \( t_{k+1} \)) during which the MUs can even change their behaviours from \( MU_{i} \) to \( MU_{j} \), wherein \( i \) and \( j \) \( \in \{1, 2, 3, 4, 5\} \). Thus, during the sifting phase, the necessity of calculating the trust values is neglected. This reduces the computational complexity. The following algorithm performs the sifting phase of the SETM method.

5.2. Evaluation Phase. This evaluation process is conducted only for SUs categorised as random or genuine. The evaluation of trust is based on the weighted average of four different types of trust. Past events include trust (\( T_{PE} \)), participation trust (\( T_{P} \)), reputation trust (\( T_{RQ} \)), and reliability trust (\( T_{RL} \)).

(i) Past event trust (\( T_{PE} \)): the past event trust is evaluated using Equation (10),

\[
T_{PE} = T(t_{s}(j - 1)) + \sum_{j=0}^{k} (-1)^{SU_{i}^{r}(t_{s})} + FCO(t_{s})
\]

Table 1: Record table at FC.

<table>
<thead>
<tr>
<th>Time slots/SU’s</th>
<th>SU₁</th>
<th>SU₂</th>
<th>SU₃</th>
<th>SU₄</th>
<th>SU₅</th>
<th>SU₆</th>
<th>SU₇</th>
<th>SU₈</th>
<th>FCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₀</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>t₁</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>t₂</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>tₖ</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
where $T(ts(j - 1))$ is the initial trust or cumulative trust of the previous time slots. $SU_j^i(ts_j)$ represents the spectrum sensing report of $SU_j^i$ over the timeslot $ts_j$. $FCO(ts_j)$ represents the final decision of the FC at time slot $j$.

(ii) Registry trust ($T_{RG}$) is a measure of the activity of the SU as selfish SUs may skip their participation in spectrum sensing to save their energy. The participation trust can be evaluated by considering the ratio of the total number of active participations ($W(i)$) to the total number of time slots. Mathematically,

$$T_{RG} = \frac{W(i)}{k} \quad (11)$$

(iii) Requite trust: this provides the reward to the genuine SU or punishment to the malicious SU. Equation (12) describes the evaluation of the requisite trust as,

$$T_{Rti} = \frac{F(i)}{k} \quad (12)$$

where $F(i)$ represents the number of falsely reported decisions by the $SU_i$, and $k$ is the number of timeslots.

(iv) Reliability trust: reliability is the quality of being trustworthy on a consistent basis. Thus, the estimation of this trust will enable us to know to what extent the SU is consistently genuine. Additionally, the reliability of MU would change often. Consistency can be measured by summing up the squared deviation of each value in any distribution divided by the number of values in the distribution. Equation (13) lists the expression used to quantify consistency

$$T_{RLi} = 1 - \sqrt{\frac{\sum (T_{oi} - T_{oj})^2}{k - 1}} \quad (13)$$

where $T_{oi} = T(i)/T(i) + F(i) + 2$ and $T_{oj} = \sum T_{oi}/k$. $T_{oi}$ is the trust considered based on the beta distribution, and $T_{oj}$ is the summation of all the individual trusts considered over the time slot $k$.

After the calculation of all the above trust indices, the weighted combination of trust is estimated based on Equation (14)

$$T_{tot} = T_{PFL} W_1 + T_{PL} W_2 + T_{RTi} W_3 + T_{RLi} W_4 \quad (14)$$

$$T_{tot} = T_{PFL} w_1 + T_{PL} w_2 + T_{RT} w_3 + T_{RL} w_4$$

where $w_1, w_2, w_3, w_4$ are the weights, and $\sum w_i = 1$.

6. Results and Discussions

The efficacy and expediency of a trust management system can be evaluated based on performance augmentation parameters, such as the low-false alarm and low misdetection rates, low overhead, and high-sensing performance. The complexity analysis would also prove the appropriateness of the proposed system [13].
6.1. Performance Augmentation. In this section, the efficacy and expediency of the proposed SETM algorithm have been discussed based on the consideration of the variation of the probability of false alarms in conjunction with the trust value and threshold. The simulations were executed initially with ten SUs with MATLAB (2014, Math Works, Natick, MA, USA). The SUs were assumed to be randomly cited around the PU whose spectrum had to be sensed. The simulation parameters for evaluation are given in Table 2.

During the sifting phase of the SETM algorithm, the sensing reports from all the SUs are given to the FCO within the prescribed time slot. The FCO identifies the different types of MUs just by analysing the sensing outputs of SUs over the given time slot with its final decision. The FCO sorts the SU as always yes MU, if the sensed output of the SU is “1” for all the time slots. The SU is sorted as always no MU, if the sensed output of the SU is “0” for the prescribed timeslot. Further, if the SU sensing output alternates for every time slot, then it is sorted as alternator MU and for any of the sensing time the SU does not sense—then the SU is sorted as selfish MU. However, if any SU is not sorted as MU during the sifting phase, then to prove the honesty of SU, the evaluation phase is conducted. During this phase, five types of trust such as past events, registries, requites, reliabilities, and cumulative trusts are evaluated. Depending on the value of the cumulative trust, the SU is classified as a random MU or as a genuine SU.

Table 3 provides an insight on the variation of past event, requisite, reliability, and cumulative trusts, with a noted increase in the percentage of malicious users. The registry trust is considered to be “1” on the assumption that the SU provides all the sensing results to the FCO within the given time slot. Further, the cumulative trust is calculated by considering the weights \( w_1 = 0.1, w_2 = 0.2, w_3 = 0.3, \) and \( w_4 = 0.4 \), as described by Equation (14). Specifically, when a SU reports falsely only once (10%), the cumulative trust value is equal to 1.67, when SU reports falsely more than once (20% to 90%) the cumulative trust decreases from the value of 1.434 to 0.264. The inference is that as the maliciousness of a SU increases, the trust value decreases. Thus, the cumulative trust can be used as a performance metric to prove whether the SU is genuine or malicious. Figure 2 shows the variation of trust as a function of the percentage of malicious behaviour in each SU, based on the use of the SETM algorithm. Specifically, for every 10% increase in malicious behavior of a SU, there is a decrease of approximately 15% in the value of cumulative trust.

Theoretically, the threshold for finding the MU is calculated by setting the variance to unity (\( \nu = 1 \)) and the probability of the false alarm to 0.1 (\( P_f = 0.1 \)) in Equation (6). The theoretical threshold value is 1.57; signifying that for the trust value of above 1.57, the SU is genuine and below that it is malicious. From Table 2, it is clear that the SU having a trust value of 1.67 and above, it is considered as genuine SU else malicious. Thus, the evaluated threshold and theoretical threshold more or less satisfies the criteria of finding the MU. Figure 3 shows the variation of the threshold as a function of the probability of the false alarm (\( P_f \)). As the probability of the false alarm increases, the threshold value decreases.

The variations of different values of trust at different timeslots are shown in Figures 4, 5, 6, and 7. The variation of past event trusts (\( T_{PE} \)) at different timeslots for different types of SUs is shown in Figure 4. The genuine SUs exhibit a \( T_{PE} \) variation between the upper and lower thresholds, where the upper threshold indicates that the respective SU is 100% genuine (having all sensing outputs correct in the previous time slots) and lower threshold indicates that the SU is 90% genuine (has reported one sensing result falsely). Thus, the past event trust of any SU fluctuates between the two thresholds (10.5 to 8.5) cannot be malicious. Another inference is that a selfish SU (MU) exhibits a trust variation between alternate timeslots. As a selfish SU does not participate in all the sensing rounds for a particular interval of time for saving its resources, the \( T_{PE} \) never reaches the upper threshold value. Also, the trust value (\( T_{PE} \)) for a random attacker (MU) lies at values which are much lower than the lower threshold, specifically below 8.5. Thus, just by evaluating \( T_{PE} \), the three types of SUs can be identified.

### Algorithm 2

Input: \( T(t_s(j - 1)), SU_j'(t_s), FCO(t_s), W(i), k, F(i), w_1 \) to \( w_4 \)

Output: Cumulative trust calculation and inference about SU

1. Calculate the past events trust using equation (10)
2. Calculate the participation trust using equation (11)
3. Calculate the requisite trust using equation (12)
4. Calculate the reliability trust using equation (13)
5. Find the cumulative Trust (\( T_{tot} \)) using equation (14)
6. If \( T_{tot} > \tau \), then SU_i = Genuine else SU_i = Random malicious attacker

### Table 2: Simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Depiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of SUs</td>
<td>10</td>
</tr>
<tr>
<td>Timing window</td>
<td>( 10 (t_{s0}, \ldots, t_s) )</td>
</tr>
<tr>
<td>Participation count (K)</td>
<td>10</td>
</tr>
<tr>
<td>Percentage of false reports</td>
<td>0.1 to 0.9</td>
</tr>
<tr>
<td>Weights for cumulative trust</td>
<td>( w_1 = 0.1, w_2 = 0.2, w_3 = 0.3, w_4 = 0.4 )</td>
</tr>
</tbody>
</table>
The identification of selfish users is also based on the calculation of the requisite trust and registry trust. Figure 5 shows the variation of the requisite trust \( T_{RQ} \) as a function of time for a genuine SU and a selfish SU. \( T_{RQ} \) increases consistently with respect to time. However, the \( T_{RQ} \) of selfish users does not yield a consistent increase. Furthermore, there is a dip of about 10% in the value of the \( T_{RQ} \) compared with the genuine SU response. Figure 6 shows the variation of the registry trust \( T_{RG} \) with the time slot. As time increases, the \( T_{RG} \) value for the genuine SU stays at a constant value, but for

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Percentage of malicious behaviour</th>
<th>Past event trust</th>
<th>Registry trust</th>
<th>Requisite trust</th>
<th>Reliability trust</th>
<th>Cumulative trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>10%</td>
<td>8.5</td>
<td>1</td>
<td>0.9</td>
<td>0.90</td>
<td>1.676</td>
</tr>
<tr>
<td>2.</td>
<td>20%</td>
<td>6.5</td>
<td>1</td>
<td>0.8</td>
<td>0.86</td>
<td>1.434</td>
</tr>
<tr>
<td>3.</td>
<td>30%</td>
<td>4.5</td>
<td>1</td>
<td>0.7</td>
<td>0.84</td>
<td>1.196</td>
</tr>
<tr>
<td>4.</td>
<td>40%</td>
<td>2.5</td>
<td>1</td>
<td>0.6</td>
<td>0.83</td>
<td>0.958</td>
</tr>
<tr>
<td>5.</td>
<td>50%</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>0.83</td>
<td>0.716</td>
</tr>
<tr>
<td>6.</td>
<td>60%</td>
<td>-1.5</td>
<td>1</td>
<td>0.4</td>
<td>0.83</td>
<td>0.502</td>
</tr>
<tr>
<td>7.</td>
<td>70%</td>
<td>-3.5</td>
<td>1</td>
<td>0.3</td>
<td>0.84</td>
<td>0.176</td>
</tr>
<tr>
<td>8.</td>
<td>80%</td>
<td>-5.5</td>
<td>1</td>
<td>0.2</td>
<td>0.86</td>
<td>-0.046</td>
</tr>
<tr>
<td>9.</td>
<td>90%</td>
<td>-7.5</td>
<td>1</td>
<td>0.1</td>
<td>0.90</td>
<td>-0.264</td>
</tr>
</tbody>
</table>

**Figure 2:** Plot of cumulative trust with respect to malicious behaviour of SU. The dots represent the value of cumulative trust for 10 to 70% variation in malicious behaviour.

**Figure 3:** Plot of threshold with respect to probability of false alarm. The dots represent the value of threshold for 0.1 to 0.5 variation of \( P_{fa} \).
the selfish SU, the TRG varies with a maximum dip of about 33\% from the constant value of genuine SU.

Reliability measures the consistency of a node. Figure 7 shows the variation of the reliability trust ($T_{RL}$) as a function of time. For the MU, the TRL never reaches the value of unity, but for genuine SU, the value stays at a constant value for most of the time. Specifically, the $T_{RL}$ value for genuine SU never falls below 90\%, but for MU, the maximum value is only 86\%. Thus, by comparing $T_{RL}$, the genuine SU and MU can be identified.

6.2. Complexity Analysis. This section investigates the appropriateness of the proposed SETM algorithm in terms of message, storage, time, and computational complexities.

(i) Message complexity (MC): defined as the number of spectrum sensing outputs required by the FC to formulate the final decision. The majority rule is used by the FC to calculate the final output. The number of SUs considered for the timeslot $t_{s0}, \ldots, t_{s9}$ is “S.” Therefore, the message complexity would be $S$

$$MC = S$$  \hspace{1cm} (15)

Storage complexity (SC): defined as the amount of memory needed for the storage of the trust value. Let “$r$” be the size of the register in the register table of the FC for storing the trust value. With regard to
the SETM algorithm, only the trust values of random MU, selfish MU, and genuine SU, need to be stored. Thus, the total storage complexity would be \((S - M)(r)\), where \(M\) is the number of malicious users filtered in the first phase of the SETM algorithm.

\[
SC = (S - M)(r)
\]  

(ii) Time complexity (TC): the timeslots needed by the FC to receive the spectrum sensed output of the SUs. Let \(t_{\text{sam}}\) be the predefined sampling period needed by the FC.

\[
TC = t_{\text{sam}}
\]

(iii) Computational complexity (CC): it is a measure that evaluates the maximum number of times the SETM is executed for the identification of malicious users. SETM has two phases: the first phase has a complexity of \(4 \times O(S) \sim O(S)\). The second phase also has a complexity of \(O(S)\), where \(S\) is the total number of SUs. \(O(S)\) represents the linear complexity.

\[
CC = O(S)
\]

The storage complexity is needed only for the genuine and random attacker SU, and not for other types of MU’s.

Tables 4 and 5 give the parameters and complexity measured for the SETM algorithm. The message complexity is the same as the number of SUs.
Institutions declare no competing interests.

As future work, the analysis of trusts can be extended by considering more than one PU.

Random attackers only need to be stored, thus lowering the overhead. As future work, the analysis of trusts can be extended by considering more than one PU.

Thus, the storage requirement and overhead are minimised. Further, the time complexity purely depends on the sampling period of CR (few milliseconds). Also, the computation complexity is the least complexity of any algorithm that is linear complexity [14]. Thus, as the computational complexity is minimum, the SETM algorithm proves to be a low-reckoning trust management system.

### 7. Conclusion

In this study, the SETM algorithm was proposed to combat SSDF attacks based on the identification of all types of malicious users that contribute to these attacks. In the first phase, the four types of SSDF attackers (always yes, always no, selfish, and alternator) were determined without the need to evaluate all the trusts. The second phase of the SETM algorithm evaluates the trust to differentiate random and genuine SUs. Thus, our method eliminates the need to evaluate the trusts for all SUs, thereby providing a low-reckoning trust-based system. Additionally, the trust values for genuine and random attackers only need to be stored, thus lowering the overhead. As future work, the analysis of trusts can be extended by considering more than one PU.

### Data Availability

The authors confirm that the data supporting the findings of this study are available within the article.

### Conflicts of Interest

The authors of this manuscript and St. Joseph’s Group of Institutions declare no competing interests.

### References


