Simple Feedback Driven Accuracy Based Reputition Mechanism for IoV

Rohan Dahiya, Frank Jiang, *DontKnow, IEEE,* and Robin Doss, *Senior Member, IEEE*

***Abstract*—The abstract goes here.**

***Index Terms*—IoV, IoT, Reputition System, Trust Management, VANET.**

I. Introduction

THIS demo file is intended to serve as a "starter file" for IEEE journal papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds

August 26, 2015

1) Subsubsection Heading Here: Subsubsection text here.

II. Related Works

A. VANETs and IoV

TODO: brief introduction to VANETS and IoV. also 5G V2X article.

B. Attacks in IoV

A survey of various attacks and detection mechanisms on VANETs and IoVs has been done by Sakiz et al. [1]. A brief description of the various attacks described is as follows.

- **Sybil Attack** A node pretends to have more then one identity.
- **DoS Attack** A Denial of Service or Distributed Denial of service attack aims to render the service unavailable by means of jamming, flooding etc.
- **Blackhole Attack** A node sends false routing information to make all other nodes try to route their packets through it, the packets are dropped.
- **Wormhole Attack** Similar to black-hole attack, in a wormhole attack, two compromised nodes forward packets between each other after encapsulating them therefore the hop count is not affected. This makes these two nodes appear as the best route to send any packets.
- **Bogus Information Attack** A type of soft attack where a malicious node sends false information in the network.
- **Replay Attack** An attacker replays a message that was sent earlier out of context. Unlikely with the use of timestamps.

R. Dahiya is a visiting researcher at the School of Info Technology, Deakin University, Burwood, VIC 3125, Australia, and a student at Vellore Institute of Technology, Vellore, TN 632014, India. e-mail: rohandahiya@outlook.in

F. Jiang and R. Doss are with School of Info Technology, Deakin University, Geelong, VIC 3220, Australia

Manuscript received Month 19, 2020; revised Month 26, 2020.

The bogus information attack mentioned above is based on the fact that vehicles in an IoV share information among each other and use that information in various protocols. In this attack a malicious node disseminates false information with the aim of manipulating the behaviour of other nodes, this affect is increased if the attacker is moving around swiftly[2]. Various variation of this attack have been studied[1] such as False Position Information[3], Sensor Tampering, Illusion Attack[4] and GPS Spoofing/Tunnel Attack[5]. This can be an issue as many IoV protocols rely on cooperativeness of vehicles to function. Various methods have been proposed for the detection of false messages as described in the following subsection (II-C). These can help a vehicle identify which messages to ignore and the results of these methods could further be used to detect and purge attackers from the network for which voting, evaluation and reputation based mechanisms have been suggested[6](*Local Eviction of Attackers by Voting Evaluators*)[7].

C. False Message Detection

Lo et al. [4] propose a mechanism called *Plausibility Validation Network (PVN)* which is capable of checking the data received from sensors, or other vehicles, and validating it. It includes a plausibility network and a rule database. For each category of messages the various rules are used to detect information that logically must be false based on known truths. Another message filtering model is proposed by Kim et al. [7]. The model includes a threshold curve and a certainty of event (CoE) curve. The CoE, which indicates the confidence level of a received message, is calculated by combining the data from various sources such as local sensors, RSUs and reputation mechanism. The solution relies on honest majority. The threshold curve shows the insensitivity of the driver with respect to the distance to the event. Sensitivity and the distance to the event are inversely proportional. Therefore, while the threshold value is decreasing, the CoE value keeps increasing and, if it exceeds the threshold value which is assigned according to the application, the driver is warned with an alert message. The paper suggested the use of a rudimentary reputation system as an input factor for the CoE score among 5 other factors.

D. Trust Management and Reputation

TODO: just to expand on this paper -¿[8]. and 2 others.

III. Proposed Methods

We Describe our proposed system for Reputation Score generation in this section as well as propose methods of

sharing feedback to the network and usage of the scores. The system is based on using an estimate of the utility of a node's previous messages as it's reputation.
We divide the section into three subsections ('Feedback Generation and Collection', 'Score Generation' and 'Usage'). The first subsection details how the feedback is created and propagated and collected in the network, in the second subsection methods of score generation are described and the third sub-section details the proposed method for the distribution/application of the scores generated. It is assumed that the RSUs can communicate with each other and therefore can maintain shared storage in the from of IPFS etc. Processing tasks can also be distributed among the RSUs (TODO add citation?), however the details of it are beyond the scope of this article. For simplicity the network of RSUs is referred to as a single unit henceforth in this section.

A. Feedback Generation and Collection

Nodes in an IoV may be able to classify some of the information they receive as true or false either by means of self observation or context aware deduction using primarily data centric methods of false information detection such as the those proposed by Kim et al.[7] and Lo et al.[4], as previously mentioned in Section II-C. If a message is determined to be true or false by a receiving node, it can share feedback on it with the other nodes of the network. The usage of this feedback is detailed in subsequent subsections. In the proposed system feedback is shared primarily in the form of a report comprising of 4 elements: the reportee's id (or address), the reporter's id (or address), the message id (or a unique message identifier), and a boolean value used to indicate if the message being reported on was found to be true or false by the reporter.

1) Feedback Generation by Nodes: The feedback exchange in the network is triggered by the following events:

- **Message Classified:** Whenever the truthfulness of a received message is determined, a report is created and shared on the network.
- **Message received from a sender for the first time:** In order to bring new nodes joining a network up to par on the feedback on nodes with the rest of the network, whenever a node a receives a message from a node b on which it has no feedback information, it (a) sends a "Request for Reports" message containing only b's id.
- **Request for Reports Recieved:** When a Request for Reports on any node b is receieved by some node a, it sends a "Reports Dump" message containing all the reports it has sent on b that it contains in storage.

2) Feedback Collection at RSU: Three lists of messages are maintained: Current Scope, Staged Scope, Archived Scope. All reports that are recieved are put into one of two baskets:

1) **Current Basket**: if the message that the report is on is not in Staged Scope or Archived Scope, the message is added to the Current Scope and the report is collected in the Current Basket.
2) **Staged Basket**: Only reports on messages that are listed in the staged scope are collected in the staged basket.

If a report is recieved on a message that is in the Archived Scope, it is ignored. Each basket stores the reports that are inserted in it and it's implementation can include variables to maintain some metadata on the reports to facilitate the score generation process which is detailed later. At regular intervals of time, which roughly equals the average amount of time between a node sending a message a node receiving a report on that message, a "Stage Shift" event is triggered. When Stage shift occurs the elements of the staged scope are added to archived scope. Secondary Scores and Message Truthvalues are calculated based on the contents of the Staged Basket before it is emptied, this process is elaborated in following subsections. These Truthvalues are then inserted into Logs. After this the contents of the Current Basket are shifted to the Staged Basket before it is emptied. Corresponding operations are performed on the Staged and Current Scope lists. The logs for each vehicle contain the calculated truthvalues of messages sent by it, and other counters that are used to calculate the Primary Scores for it with optimal complexity. It can be summarised that the Archived Scope is the list of messages for which the truthvalues have been calculated and inserted into the logs, the Staged Scope is the list of messages for which the reports are still being collected and that will be processed and inserted into logs at the next stage shift event, and the current scope is the expanding list of messages that will form the fixed message list of staged scope at the next stage shift event.

3) RSU Results Broadcasting and incorporation at Nodes: As mentioned previously, the Logs at the RSU contain the calculated truth-values of messages which are used to generate the Primary Scores, the calculation of these truth-values and Primary Scores is detailed in the subsequent subsection. In order to share it's results the RSU calculates the Primary Score of each node for various window sizes i.e [10,50,250,1250] or such and sends these (with other metadata) along with the list of blacklisted nodes in a message to all nodes in the network. Whenever a message from an RSU is received the node can make a copy of the blacklist to ignore reports from the blacklisted nodes, untill the blacklist is updated again and it can insert dummy messages/reports from various nodes into it's own storage by reverse calculating from the received primary scores and metadata in order to emulate the same knowledge.

B. Score Generation

We describe the method of score-generation in this subsection. All data pertaining to reports and counters in this subsection and in Figures 1 to 4 refers to the data collected in the staged basket up until the stage shift event occurs and the various score generation methods are called, unless otherwise specified to refer to data in Logs. In order to generate an estimate of the overall accuracy of all the information shared by a node, the system calculates a truth-value for each message sent by that node based on all the feedback on that message, then the mean of the truth-values of a set of messages is taken. This estimate of the overall accuracy of a node is referred to as the Primary Score or the Reputation Score interchangeably in

this article. Since the feedback may be deliberately falsified as well, it is first necessary to exclude the feedback from nodes that may be malicious. In order to distinguish nodes with false feedback another parameter is used which is referred to as the "Secondary Score" for a node. The following sub-sub-sections describe the methods for ultimately generating the Primary Score for each node.

1) Secondary Score Generation: When ever a new report is received by the RSU, counters corresponding to the number of positive and negative reports on a node and the number of positive and negative reports from a node on another node can be incremented. These counters can then be utilised to quickly calculate each node's median implied reputation score (MI Score), which is defined as the median of all the implied reputation scores for that node, where an implied reputation score of a node is the fraction of positive reports to total reports by another node. See Fig. 1, Alg. 2. By assuming the MI score for each node as a benchmark, the deviation of the scores implied by a node's reports from the respective MI scores is utilised as a measure of the utility of it's reports. This deviation is measured in terms of Mean Squared Deviation per report and is termed as the Secondary Score of the Node. It is calculated for each node as described in Fig. 1 Alg. 2.

2) Blacklist Generation: The Secondary Scores are a measure of abnormality on the reports of a node. This measure may be used to scale down the effect of it's reports during the Primary Score generation, or more simply it may be used to blacklist nodes who's Secondary Score crosses a certain threshold. The latter approach was pursued by us during experimentation where a simple heuristic method of determining a dynamic threshold based on the distribution of the secondary scores was adopted. With the assumption of honest majority the median of secondary scores must not be more than that of a dishonest node with abnormally lower utility of reports. Therefore the threshold value was defined as a function of the median of all Secondary Scores and the median absolute deviation around that median. The process of the generation of the blacklist is as described in Fig. 2 Alg. 3

3) Primary Score Generation: The counters utilised during secondary score calculation may also be used to calculate estimates of accuracy for each node in constant time as described in Fig. 3 - Alg. 4. This estimation is referred to as RAW Score and serves as a baseline during experiments, in itself the RAW score is highly inaccurate however it is quick to calculate.

With all the reports on each message indicating it to be either true or false, the mean of these reports can be calculated and treated as a fuzzy truth-value of it. As mentioned earlier the primary score is an estimation of the accuracy of all the information shared by a node, this can be trivially calculated as the mean of the calculated truth-value of a set of messages by that node. During this calculation the reports by blacklisted nodes are ignored. The process of calculating these truth-values is as described in Fig. 4 - Alg. 5. These calculated truth-values for messages are then inserted into the logs. From these logs we generate the Primary Score which is the mean of the truth-values of a set of messages. Calculating the Primary Score over the all the messages would give a close estimate of a nodes actual accuracy assuming it remains consistent however it is not likely to reflect a change in the accuracy of a node quickly. On the other hand the Primary Score of a node over a small window of message truth-values is likely to fluctuate more on every stage shift and not be an accurate estimate of the behaviour of a node however it would reflect a

Fig. 1: Calculation of Secondary Scores

Algorithm 1 Calculate Median Implied Scores

INPUT:

1) N: Set of all nodes on which at least one report was received
2) $Count_{i,j}^{Total}$: Count of Total Reports on node i by node j $\forall i, j \in N$
3) $Count_{i,j}^{Positive}$: Count of Positive Reports on node i by node j $\forall i, j \in N$

OUTPUT: $Score_n^{MIS}$: Median Implied Score of node n $\forall n \in N$

1: **for all** $i \in N$ **do**
2: $impliedScore_j \leftarrow Count_{i,j}^{Positive}/Count_{i,j}^{Total}$ $\forall j \in N - \{i\}$
3: $Score_i^{MIS} \leftarrow$ median($impliedScore_j \ \forall \ j \in N - \{i\}$)
4: **end for**
5: **return** $Score_i^{MIS} \ \forall i \in N$

Complexity: $\mathcal{O}(n^2)$.

Algorithm 2 Calculate Secondary Scores

INPUT:

1) N: Set of all nodes from whom at-least 1 report was received
2) $Count_{i,j}^{Total}$: Count of Total Reports on node n by node j $\forall i, j \in N$
3) $Count_{i,j}^{Positive}$: Count of Positive Reports on node n by node j $\forall i, j \in N$
4) $Score_n^{MI}$: MI Score of node n $\forall n \in N$

OUTPUT: $Score_n^{Secondary}$: Secondary Score of node n $\forall n \in N$

for all $i \in N$ **do**
  $TSE \leftarrow 0$
  $TotalReports \leftarrow 0$
  for all $j \in N \mid j \neq i$ **do**
    $impliedScore \leftarrow Count_{j,i}^{Positive}/Count_{j,i}^{Total}$
    $error \leftarrow Score_j^{MI} - impliedScore$
    $TSE \leftarrow TSE + (error^2 \times Count_{j,i}^{Total})$
    $TotalReports \leftarrow TotalReports + Count_{j,i}^{Total}$
  end for
  $Score_i^{Secondary} \leftarrow \frac{TSE}{TotalReports}$
end for
return $Score_i^{Secondary} \ \forall i \in N$

Complexity: $\mathcal{O}(n^2)$.

Fig. 2: Generation of Blacklist

Algorithm 3 Generate Blacklist

INPUT:

1) N: Set of all nodes for which a secondary score exists
2) $Score_n^{Secondary}$: Secondary Score of node $n \; \forall \; n \in N$

OUTPUT: $Blacklist$: Set of all blacklisted nodes

$m \leftarrow$ median($\{Score_i^{Secondary} \; | i \in N\}$)
$ADm \leftarrow \emptyset$
for all $i \; \in \; N$ **do**
  $ADm \leftarrow ADm \cup \{|Score_i^{Secondary} - m|\}$
end for
$MADm \leftarrow$ median(ADm)
$threshold \leftarrow m + 2 \times MADm$
$Blacklist \leftarrow \emptyset$
for all $i \; \in \; N$ **do**
  if $Score_i^{Secondary} > threshold$ **then**
    $Blacklist \leftarrow Blacklist \cup \{i\}$
  end if
end for
return $Blacklist$

Complexity: $\mathcal{O}(n)$.

Fig. 3: RAW Score Calculation

Algorithm 4 Calculate RAW Scores

INPUT:

1) N: Set of all nodes on whom atleast 1 report exists
2) $Count_i^{Total}$: Count of Total Reports on node $i \; \forall i \in N$ in Staged Basket as well as Logs.
3) $Count_i^{Positive}$: Count of Positive Reports on node $i \; \forall i \in N$ in Staged Basket as well as Logs.

OUTPUT: $Score_n^{RAW}$: RAW Score of node $n \; \forall n \; \in N$

for all $i \; \in \; N$ **do**
  $Score_i^{RAW} \leftarrow (Count_i^{Positive}/Count_i^{Total})$
end for
return $Score_i^{RAW} \; \forall i \in N$

Complexity: $\mathcal{O}(n)$.

sudden change in th accuracy of a node quickly. The process of generating Primary Scores over a given window size for all nodes is described in Fig.4 - Alg.6

C. Usage

Trends in the Primary Scores of each nodes, over different window sizes, can be tracked to detect malicious behaviour such as perpetual inconsistency, drastic changes etc. besides the trivial case of being low. Likewise Secondary Scores can be monitored as well to identify and remove nodes from

Fig. 4: Primary Score Calculation

Algorithm 5 Calculate Truth-values

INPUT:

1) N: Set of all nodes
2) M_i: Set of all messages (ids) from node $i \; \forall \; i \; \in \; N$
3) $Report_j^{i,m} \; \in \; \{0, 1\}$: Alleged Validity of message m by node i in a Report by node $j \quad \forall m \; \in \; M_i | Report_j^{i,m}$exists $\forall \; i, j \; \in \; N | i \neq \; j$
4) B: Set of all blacklisted nodes

OUTPUT:

1) M_i': Set of all messages (ids) from node i for which a truth-value is calculated $\; \forall \; i \; \in \; N$
2) tv_m^n: Truth-value of message m from node $n \; \forall m \; \in \; M_n' \; \forall n \in N$

for all $n \in N$ **do**
  $M_n' \leftarrow \emptyset$
  for all $m \in M_n$ **do**
    $R \leftarrow \{Report_j^{n,m} | \; j \in N - B$ **and** $Report_j^{n,m}$exists$\}$
    if $|R| \neq 0$ **then**
      $tv_m^n \leftarrow \frac{\sum_{r \in R} r}{|R|}$
      $M_n' \leftarrow M_n' \cup m$
    end if
  end for
end for
return $tv_m^n \forall m \in M_n' \; \forall n \in N$

Complexity: $\mathcal{O}(n^2 m)$.

Algorithm 6 Calculate Primary Scores

INPUT:

1) N: Set of all nodes in Logs
2) M_i: List of all messages (ids) in Logs from node i in reverse order of arrival $\; \forall \; i \; \in \; N$
3) tv_m^n: Truth-value of message m from node $n \; \forall m \; \in \; M_n \; \forall n \in N$
4) w: Size of the window for which average truth-value needs to be calculated

OUTPUT: $Score_n^{Primary,w}$: Primary Score of Node n calculated over a window of size $w \quad \forall n \quad \in N$

for all $n \in N$ **do**
  $M' \leftarrow M_n[1 \text{ to } w]$
  $sum \leftarrow \sum_{m \in M'} tv_m^n$
  $Score_n^{Primary,w} \leftarrow sum/w$
end for
return $Score_n^{Primary,w} \; \forall n \in N$

Complexity: $\mathcal{O}(nm)$.
Complexity *for optimized implementation of logs*: $\mathcal{O}(n)$.

the network. In the process of calculating the MI Scores prior to Secondary Score Calculation (See Fig. 1 Alg. 1) the distribution of the implied scores can be analysed by

using statistical methods of multimodality detection such as Dip test[9] or Silverman's test[10], where a high degree of bimodality can be an indicator of collusion to manipulate the reputation system. These options of deeper analysis have not been further explored in this article however have been mentioned here as an indicator of possible extension of the methods. In the scope of this article the calculated truth values are used to calculate the Primary Scores at various sizes and share them with the Nodes where their utilisation can be done in a number of alternative ways. The application of the generated scores could be as a an input to the false message detection system that generated the feedback this system depends on, as some methods consider reputation of the sender as a parameter for evaluation[7]. The reputation score could also used to select vehicles that must be removed from the network or it could be used as a factor from a broader trust management system[8].

1) Usage at RSU: As previously mentioned in Section III-A, at the RSU the Primary Scores are calculated at certain window sizes for all nodes and this data along with some meta data such as the id or timestamp of the last message considered from each node is sent along with the blacklist as a message to the network. At the nodes this data of primary scores along with the metadata is used to create dummy message truth-values or reports (depending on the usage) to emulate the data that would produce these Primary Scores, This is done to reduce the amount of information that needs to be sent over the network.

2) Usage at Nodes - Disabled Node: The nodes can depend on the RSU entirely to provide the Primary Scores. Pessimistically minimum of the primary scores of a node (over different window sizes) can be taken.

Advantages: Low Computation at nodes

Disadvantages:

- Less adaptable to change in behaviour of nodes, scores remain static between RSU Stage Shifts
- System is Entirely dependant on RSU

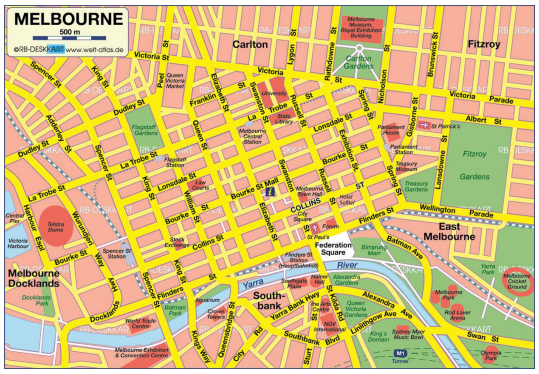
3) Usage at Nodes - Adapted RAW Score Algorithm: Reports can be used by nodes to calculate Reputation Scores by using an adaptation of the RAW Score Algorithm at the RSU (See Alg. 4) where the average of the last w' reports on a node is taken as a Primary score over a window size of w'. When a RSU Broadcast is received, for each window size w for which a primary score exists on nodes, w messages with truth-value $tv = Score^{Primary,w}$ are emulated. For each message to be emulated x number of dummy reports, whose average $\approx$ desired truth-value, are inserted into the logs, where x = average number of reports per message (this can be included as part of the metadata in the RSU broadcast). The window sizes to be used when calculating Primary Scores locally can be calculated by scaling the window sizes used at the RSU by a factor of x.

Advantages:

- More adaptable to change in behaviour of nodes, scores are dynamic between RSU Stage Shifts
- Moderate Computation at nodes

Disadvantages:

Fig. 5: Melbourne CBD Road Network



- System is Largely dependant on RSU, Estimates will become highly inaccurate over time without RSU updates

4) Usage at Nodes - Adapted Primary Score Algorithm: Reports can be used by nodes to calculate Reputation Scores by using an adaptation of the Primary Score Algorithm at the RSU (See Alg. 5 and 6) where the average of the truth-values of the last w messages on a node is taken as a Primary score over a window size of w. Messages with less than a certain number of reports on it can be kept in a buffer while more reports are received. Whenever a new report on a message is received it's truth-value is accordingly updated and so is the Primary Score over windows what include the message. When a RSU Broadcast is received, for each window size w for which a primary score exists on nodes, w dummy messages with truth-value $tv = Score^{Primary,w}$ are inserted. Reports from nodes that are blacklisted are purged and so are the reports on messages that have been evaluated by the RSU for Score Genaration. This data can be provided in the metadata of the RSU Broadcast

Advantages:

- More adaptable to change in behaviour of nodes than disabled node, scores are dynamic between RSU Stage Shifts
- System is Less dependant on RSU, Estimates will remain accurate to a high degree over time without RSU updates in the absence of malicious nodes that share false feedback.

Disadvantages:

- System would be adversely affected in the absence of RSU if malicious nodes send false feedback.
- Higher amount of computation and storage at node

IV. Experiments

The system was tested by software simulation in various scenarios. Details on the Simulation Set-up and parameters can be found in Appendix A. Three different cases in two separate environments were simulated to give a total of six scenarios. The Environments were as follows:

A. Environments

1) City: The road network of Melbourne CBD was taken with vehicles spawning at random locations, at random times,

driving to a random destinations, then going off-line. Short lifespans of individual vehicles, however the network is always densely populated with new vehicles appearing as old ones go offline.
Meant to simulate the situation of traffic in a busy city centre. See Fig.. 5

2) Highway: A flow of a fixed number of vehicles was simulated to run on a fixed route, mostly comprised of straight multi-lane roads. All nodes had larger lifespans than before. By means of overtaking and other reasons, the order of initialisation was not maintained as the order in which the nodes continued to travel.
It was meant to simulate the situation of vehicles moving on a highway for a flow of vehicles from a point to a distant point.

B. Situations

Different situations were simulated to analyse the system's performance against various attack vectors of compromising the the systems ability to produce accurate estimates. The behaviour of a regular node was programmed as follows: Overall accuracy of Sent Messages is around 90%. The node can determine the truthfulness of 60% messages it receives with an accuracy of 95%. TODO: add citation. Further details of a regular nodes behaviour can be found in appendix A.

1) Situation 0: The first situation considered was that of a few (10%) malicious nodes that send false information on the network with an accuracy of 5%. The rest of the parameters controlling their behaviour was the same as regular nodes. This situation enables us to analyse the system's performance at detecting false messages and malicious nodes in a scenario where no node is sharing falsified feedback to jeopardise the reputation system. This can then be compared with other situations to spot degradation in performance.

2) Situation 1: In the second situation, over and above the factors present in situation 0, 10% Nodes were programmed to send reports with an accuracy of 5% i.e 95% reports generated by them are falsified and they send reports on 100% messages they receive as opposed to 60%, this was implemented to observe the degradation in the accuracy of estimates with false reports being shared by nodes and the ability of the system to detect the nodes sending falsified reports.

3) Situation 2: In the third and final situation, over and above the factors present in situation 0, 20% of nodes were programmed to target 5% of nodes specifically. These colluding nodes would send falsified feedback on every message received from target nodes while sending useful reports otherwise. This was implemented to analyse the systems ability to detect convoluted methods of compromising it's ability of estimating accuracy where a large percent of malicious nodes send mostly useful feedback in order to not get blacklisted but target specific nodes, either regular or malicious, to alter their primary score.

C. Scenarios

The various permutations of environments and situations were termed as follows.

- Scenario 0: Situation 0 in City

Fig. 6: Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error in Scenario 0 (above) and Scenario 1 (below)

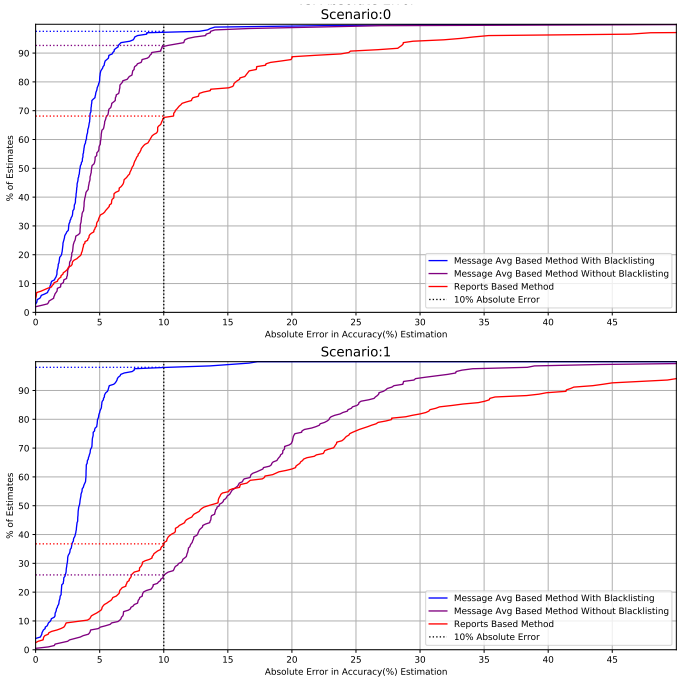
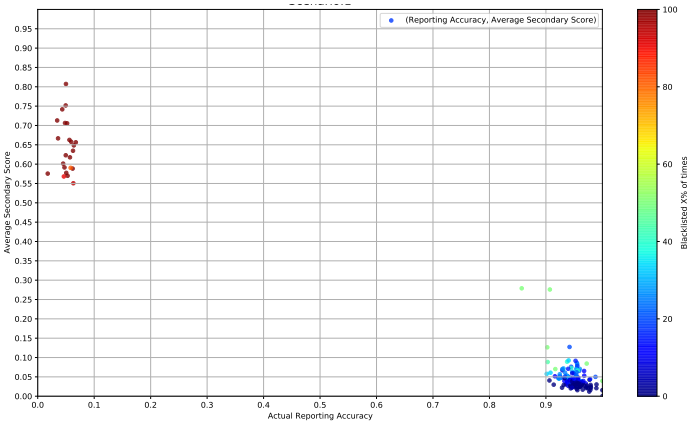


Fig. 7: Mean Secondary Score vs. Accuracy of Reports Sent, Scatter plot for Scenario 1



- Scenario 1: Situation 1 in City
- Scenario 2: Situation 2 in City
- Scenario 10: Situation 0 on Highway
- Scenario 11: Situation 1 on Highway
- Scenario 12: Situation 2 on Highway

V. Results

Various plots on the results of the simulations for each scenario are presented in Appendix. B. It can be observed from the results that in the absence of falsified reports, the system can estimate the accuracy of almost all (97%) nodes in city environment and all nodes in highway environment with less than 10 absolute error where accuracy and scores

Fig. 8: Mean Secondary Score vs. Accuracy of Reports Sent, Scatter plot for Scenario 2

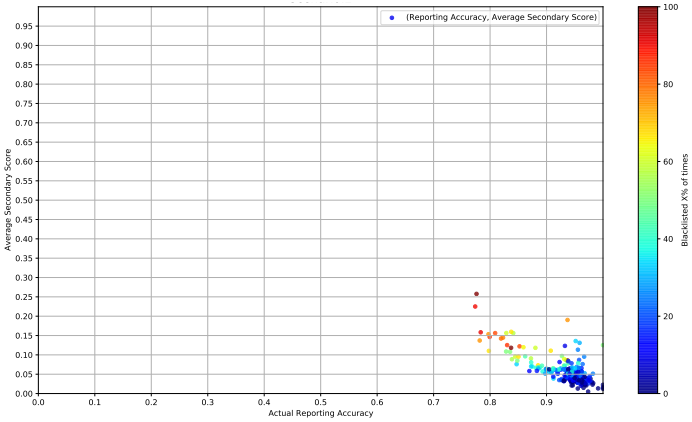
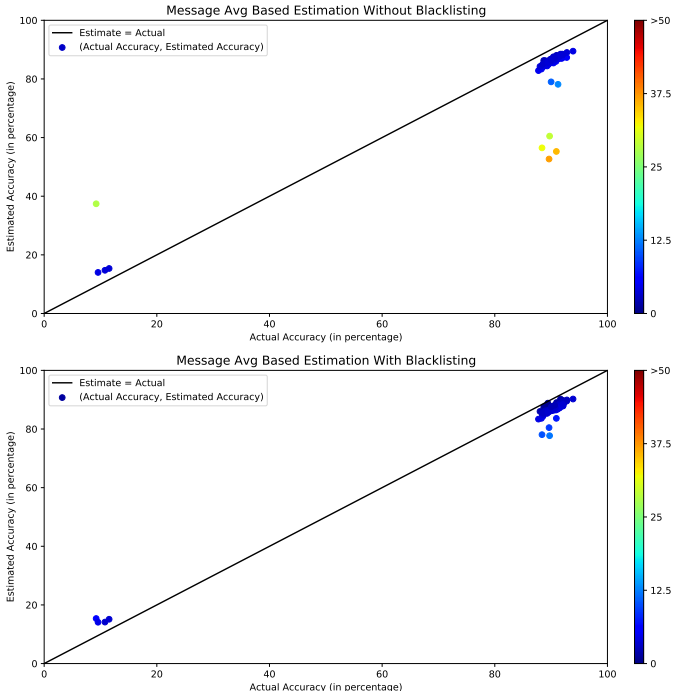


Fig. 9: Primary Score vs. Actual Accuracy - Scatter plot for Scenario 12. Without blacklisting (above) and with blacklisting (below).



were measured in percentage.

Comparing the system's performance between scenario 0 and scenario 1 (where falsified reports were being sent) shows that significant percentage of scores are highly inaccurate for scores calculated without blacklisting. In scenario 0, 92.2% of accuracy estimates without blacklisting had less then 10 absolute error (0.1 when normalised) where accuracy and scores were measured in percentage. In scenario 1 this dropped down to 25.5% of estimates. Whereas the same measure for score calculation with blacklisting remained about the same. This can be seen in Fig. 6. Similar observations can be made when comparing scenario 10 and 11. This lack of degradation of the system can be attributed to it's ability to accurately identify nodes that send false feedback all the time. This can be seen in Fig. 7, where nodes with low utility of reports were almost always blacklisted as indicated by the deep red colour. In Scenarios 2 and 12, where a more intelligent approach to sabotage the system's accuracy estimate (score) of a node was made by 20% nodes in the network that acted like regular nodes except when sending reports on 5% nodes that were the targets for the collusion, the accuracy of those target nodes was highly inaccurate as expected, this can be seen in Fig. 9 (lower) where the accuracy of 5 nodes was uniquely inaccurate. With blacklisting, however, the errors were minimal as can be observed in Fig. 9 (upper). This lack of error can again be attributed to the system's ability to accurately identify and blacklist nodes that send false feedback even some of the times. This can be observed from in Fig. 8 where nodes with slightly lower utility of reports blacklisted sometimes to most often as indicated by the light blue to red colours. It is important to note that the special collusion detection methods of multimodality detection[9][10] as explained in section III-C was not leveraged during experiments.

VI. Conclusion

T

Appendix A
Simulation Set-Up

A. Traffic Simulation

Traffic was simulated on SUMO. For city environment the road network of Melbourne CBD was generated using the osm web wizard script. Vehicle trips for this road network were created by a call to the random trips script from within the osm web wizard script. Routes for the generated random passenger trips were calculated with explicitly calling the duarouter method of SUMO. For the Highway Scenario, road network of larger Melbourne area was generated using the osm web wizard script. After this two points on the network, one in front of Deakin University, Burwood, and one in Fitzroy were chosen as starting and ending points. A route for the same was calculated using Duarouter and a flow of 50/100 vehicles was generated on that route.

B. Network/Application Simulation

Veins on OMNET++ was utilised to simulaate the network and the application. The default parameters in veins for antenna strength etc. were utilised which are based on TODO add citation.

C. Regular Nodes

- Overall accuracy of Sent Messages is around 90%.
- Can determine the truthfulness of 60% messages it receives with an accuracy of 95%.
- Sends a message on a network every 4s$\pm$ 2000ms.
- Sends a report (its evaluation of a received message, if evaluated) in 2s$\pm$ 1000ms
- Sends a request on the network for all nodes to send their evaluations of all messages from a node when a

message from a node is encountered for the first time in 1s± 500ms
- Can oblige such a request in 1s± 500ms

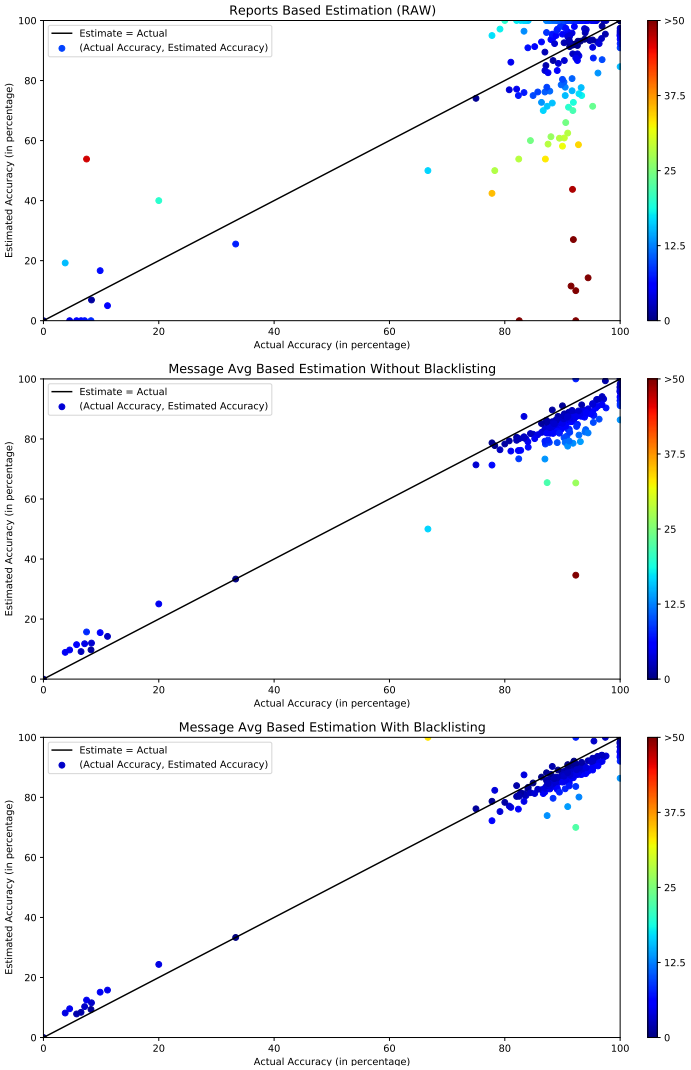
Other nodes' behaviour was different from a regular in only ways previously explained in Section IV (IV-B2 and IV-B3).

APPENDIX B
SIMULATION RESULTS

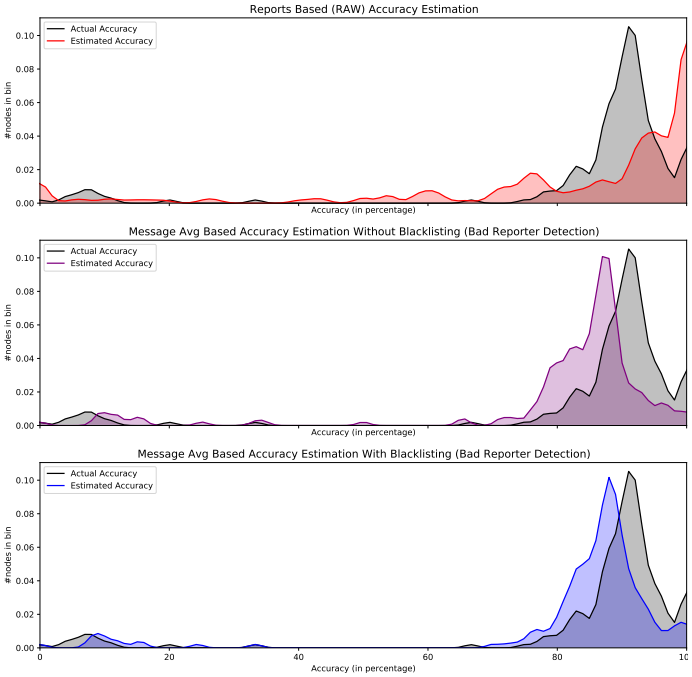
The the performance of the system in Scenario 0, 1, 2, 10, 11, and 12 is visualised in Figures B.1, B.2, B.3, B.4, B.5 and B.6 respectively. In these figures "Message Average Based Estimation with Blacklisting" refers to the proposed system, "Message Average Based Estimation without Blacklisting" refers to score calculation without blacklisting and "Reports Based Estimation" refers to the RAW Score (See Section III-B)

Fig. B.1: Graphs for Results of Scenario 0. (a) Primary Score vs. Actual Accuracy - Scatter plot. (b) Primary Score and Actual Accuracy KDE. An Indicator of closeness of estimates. (c) Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error. (d) Mean Secondary Score vs. Accuracy of Reports Sent - Scatter plot.

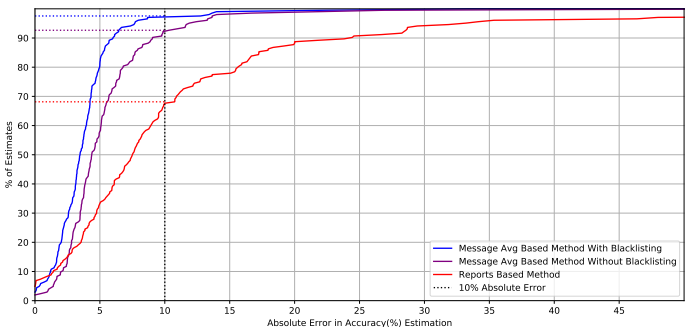
(a)



(b)



(c)



(d)

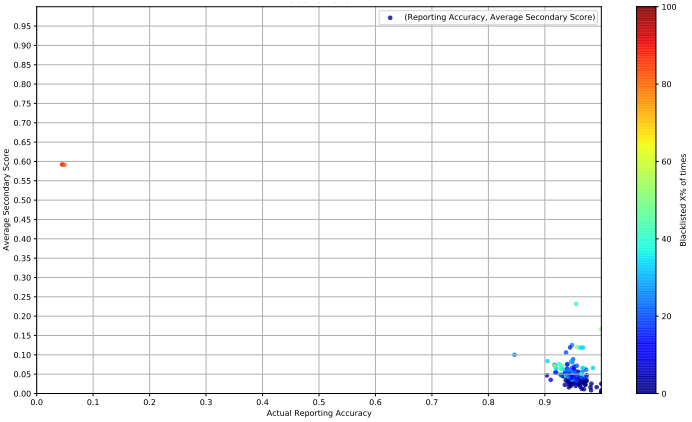
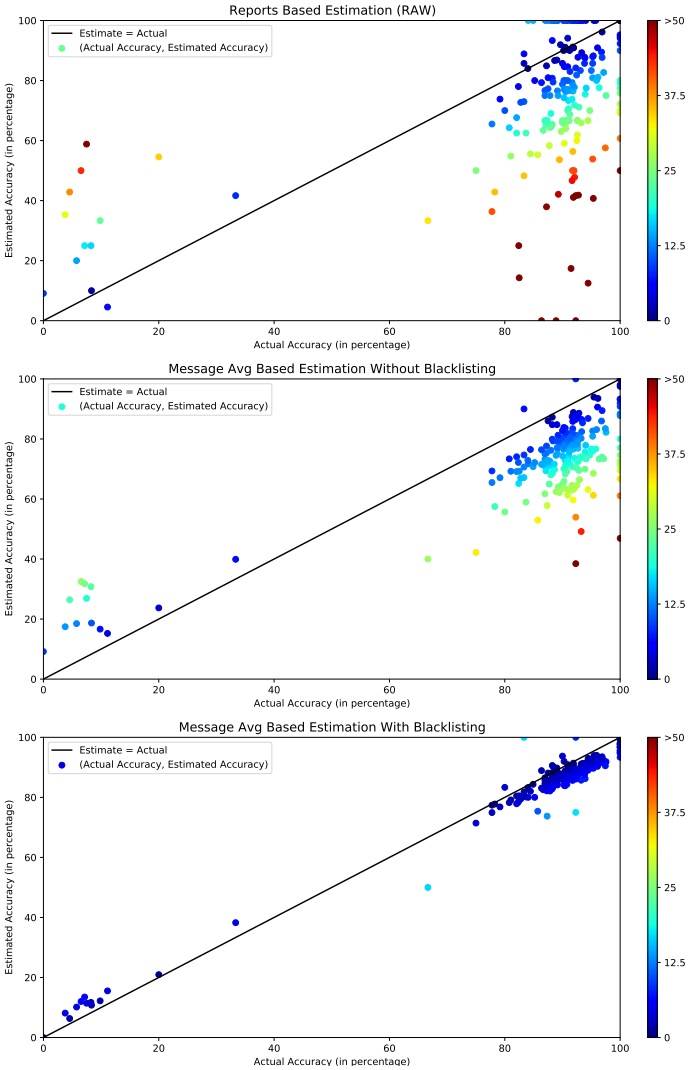
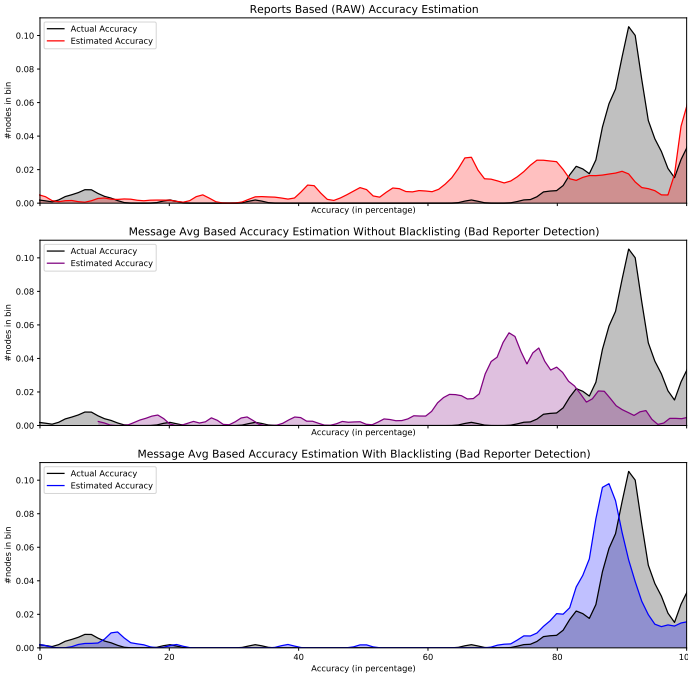


Fig. B.2: Graphs for Results of Scenario 1). (a) Primary Score vs. Actual Accuracy - Scatter plot. (b) Primary Score and Actual Accuracy KDE. An Indicator of closeness of estimates. (c) Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error. (d) Mean Secondary Score vs. Accuracy of Reports Sent - Scatter plot.

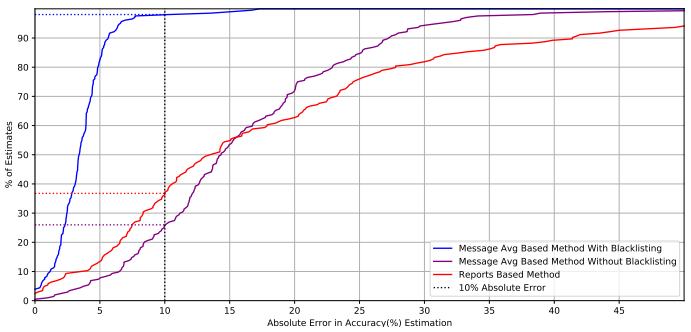
(a)



(b)



(c)



(d)

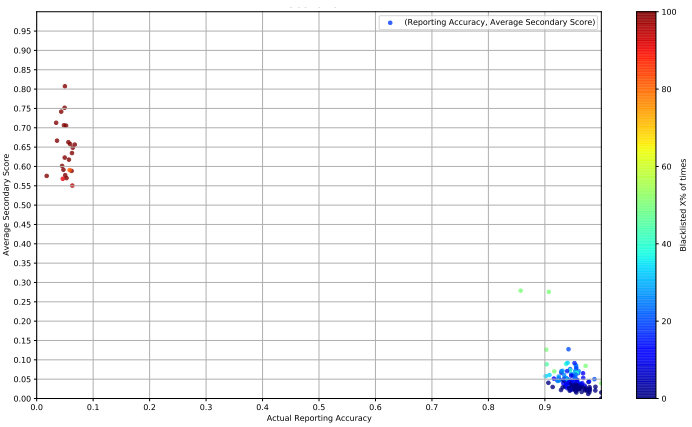


Fig. B.3: Graphs for Results of Scenario 2). (a) Primary Score vs. Actual Accuracy - Scatter plot. (b) Primary Score and Actual Accuracy KDE. An Indicator of closeness of estimates. (c) Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error. (d) Mean Secondary Score vs. Accuracy of Reports Sent - Scatter plot.

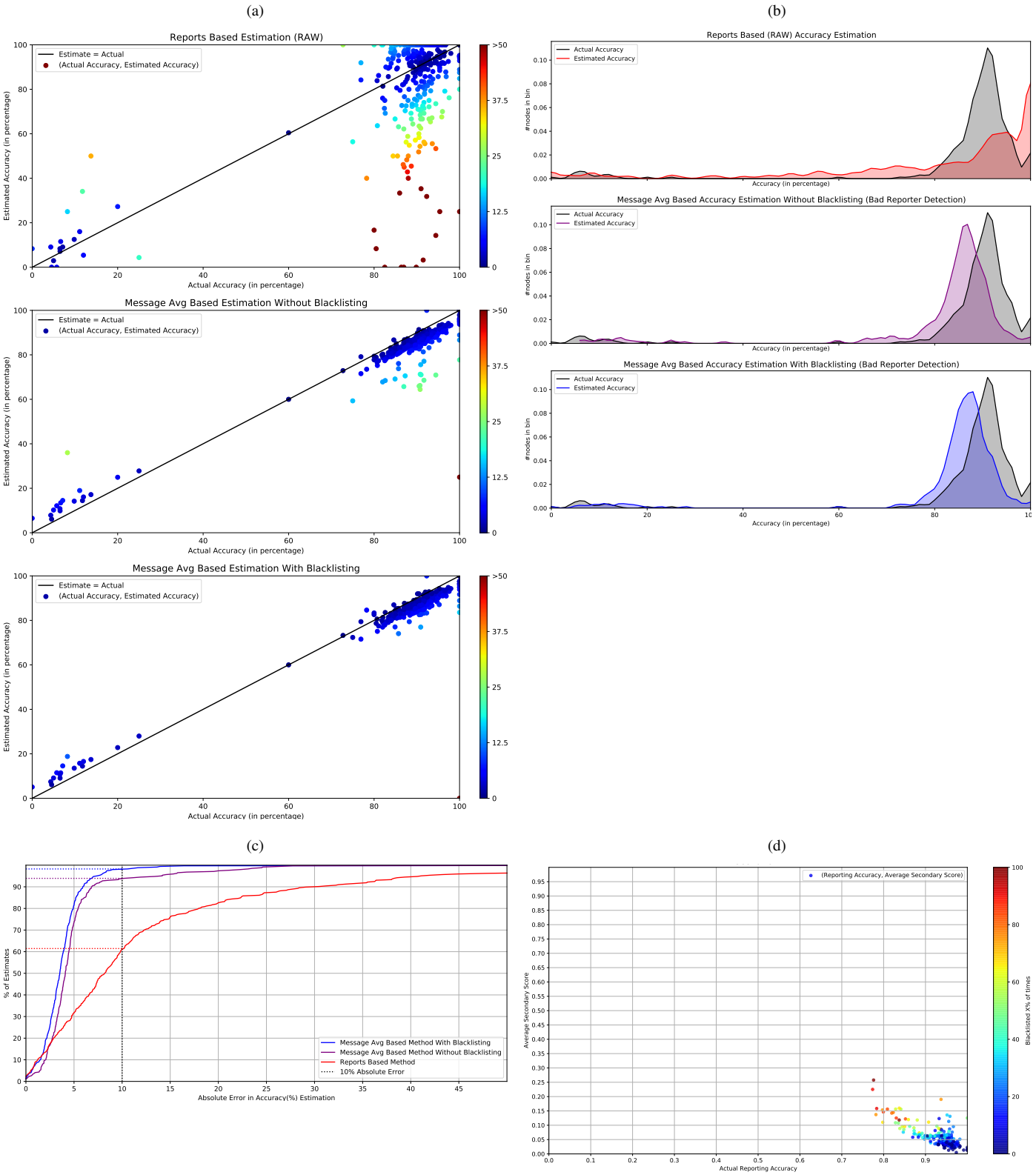
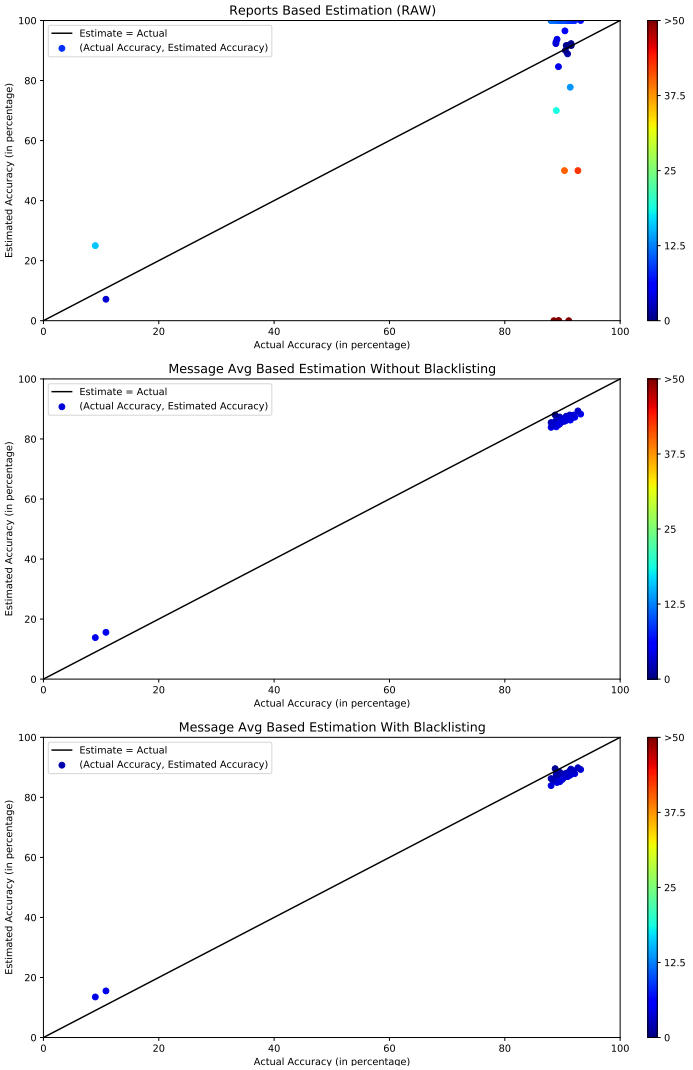
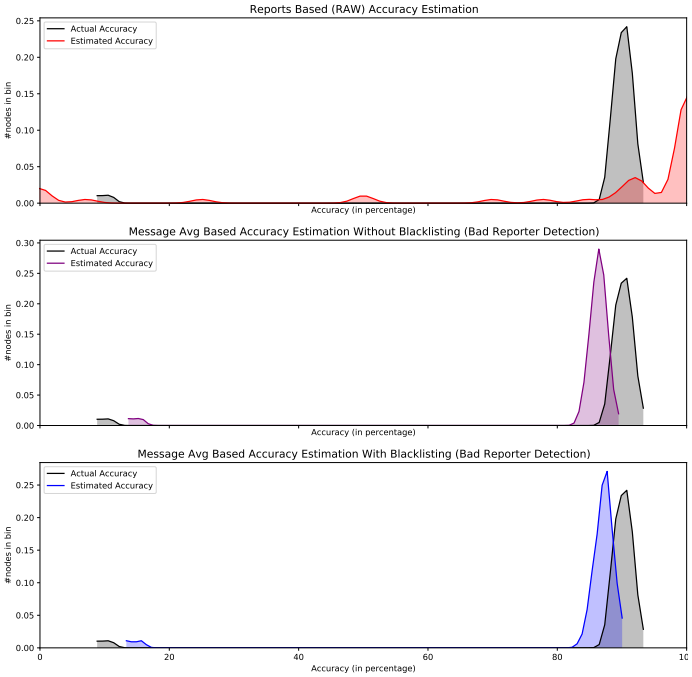


Fig. B.4: Graphs for Results of Scenario 10. (a) Primary Score vs. Actual Accuracy - Scatter plot. (b) Primary Score and Actual Accuracy KDE. An Indicator of closeness of estimates. (c) Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error. (d) Mean Secondary Score vs. Accuracy of Reports Sent - Scatter plot.

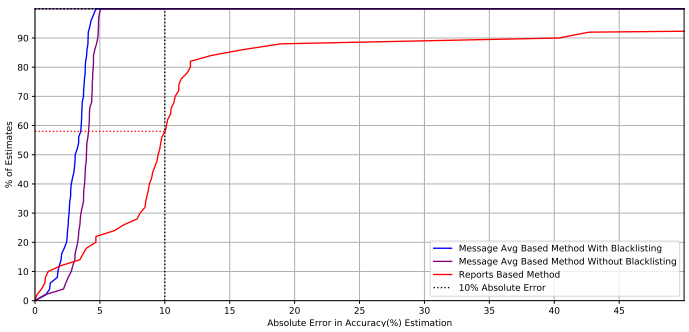
(a)



(b)



(c)



(d)

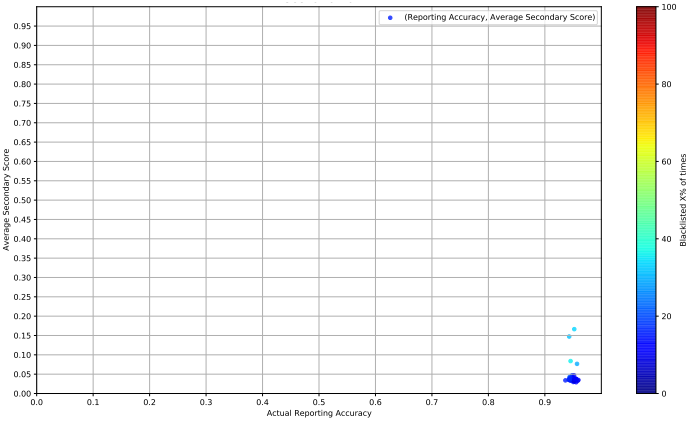


Fig. B.5: Graphs for Results of Scenario 11. (a) Primary Score vs. Actual Accuracy - Scatter plot. (b) Primary Score and Actual Accuracy KDE. An Indicator of closeness of estimates. (c) Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error. (d) Mean Secondary Score vs. Accuracy of Reports Sent - Scatter plot.

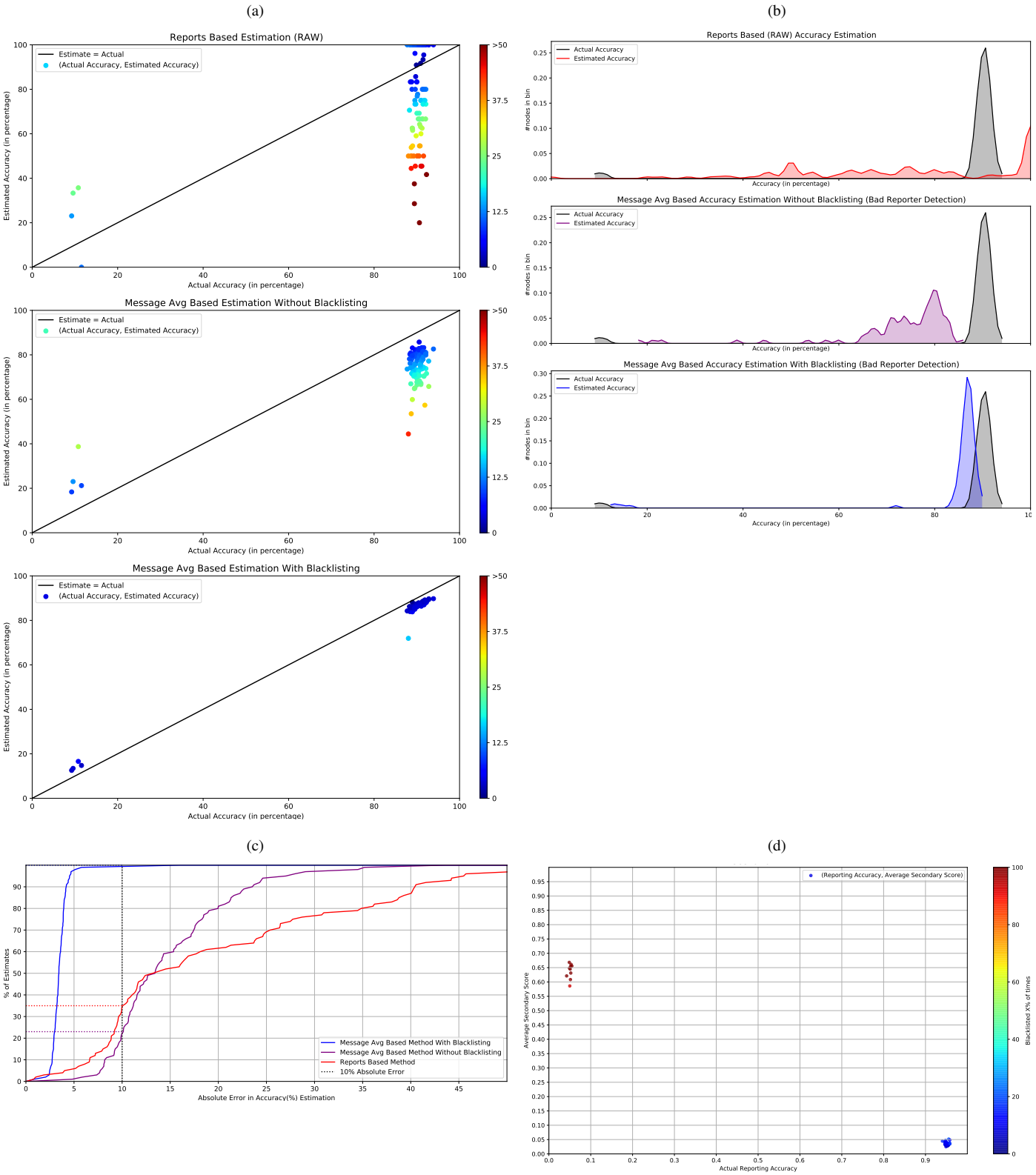
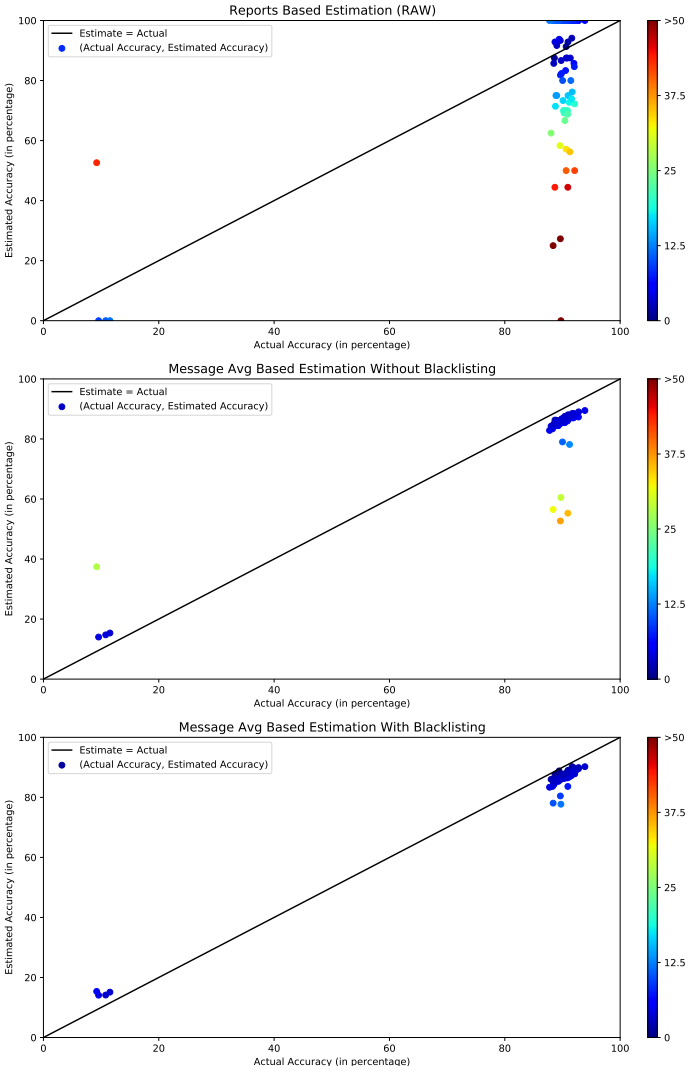
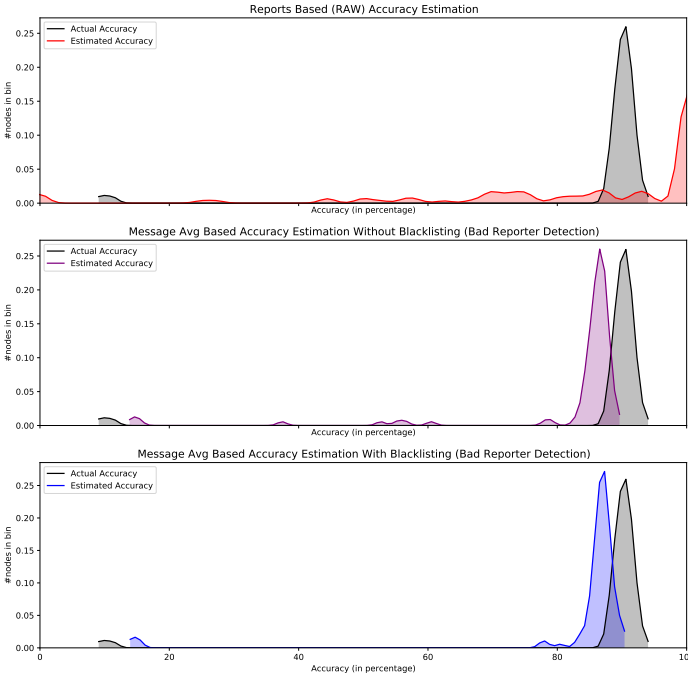


Fig. B.6: Graphs for Results of Scenario 12. (a) Primary Score vs. Actual Accuracy - Scatter plot. (b) Primary Score and Actual Accuracy KDE. An Indicator of closeness of estimates. (c) Absolute error (difference between actual accuracy of node and estimated accuracy) vs. Percentage of nodes for which there was less than 'x'% error. (d) Mean Secondary Score vs. Accuracy of Reports Sent - Scatter plot.

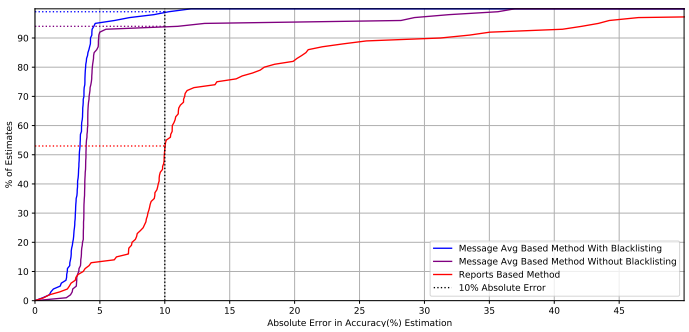
(a)



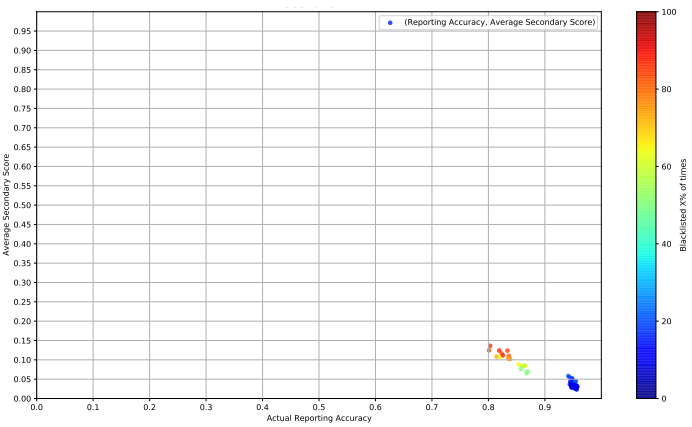
(b)



(c)



(d)



REFERENCES

[1] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov," *Ad Hoc Networks*, vol. 61, 03 2017.

[2] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, and J. Hubaux, "Fast exclusion of errant devices from vehicular networks," in *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2008, pp. 135–143.

[3] T. Leinmller and E. Schoch, "Greedy routing in highway scenarios: The impact of position faking nodes," 01 2006.

[4] N. Lo and H. Tsai, "Illusion attack on vanet applications - a message plausibility problem," in *2007 IEEE Globecom Workshops*, 2007, pp. 1–8.

[5] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, p. 3968, Jan. 2007.

[6] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1557–1568, 2007.

[7] T. Kim, A. Studer, R. Dubey, X. Zhang, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Vanet alert endorsement using multi-source filters," 01 2010, pp. 51–60.

[8] R. Iqbal, T. A. Butt, M. Afzaal, and K. Salah, "Trust management in social internet of vehicles: Factors, challenges, blockchain, and fog solutions," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, p. 1550147719825820, 2019. [Online]. Available: https://doi.org/10.1177/1550147719825820

[9] J. A. Hartigan and P. M. Hartigan, "The dip test of unimodality," *The Annals of Statistics*, vol. 13, no. 1, pp. 70–84, 1985. [Online]. Available: http://www.jstor.org/stable/2241144

[10] B. W. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 43, no. 1, pp. 97–99, 1981. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1981.tb01155.x

Rohan Dahiya Rohan Dahiya was born in New Delhi, India in 1997. He is currently pursuing the B.Tech. degree in computer science and engineering with specialisation in information security from Vellore Institute of Technology, TN, India.
He has worked as an Intern at Volon Cyber Security in the summer of 2018 and at GAP IT Services India in the summer of 2019. He is currently a Visiting Researcher at Deakin University, Burwood, VIC, Australia.

Frank Jiang Frank Jiang received the masters degree in computer science from the University of New South Wales (UNSW), Australia and the Ph.D. degree from The University of Technology Sydney, Australia. He gained three and a half years of postdoctoral research experience at UNSW. He is currently a Senior Lecturer of cyber security with the School of Info Technology Campus, Deakin University, Australia. He has published over 100 highly reputed SCI/EI indexed journal/conferences papers. His main research interests include data-driven cyber security, predictive analytics, biologically inspired learning mechanism, and its application in the complex information security systems.

Robin Doss Robin Doss (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering from the University of Madras, India, and the masters and Ph.D. degrees from the Royal Melbourne Institute of Technology (RMIT), Australia. He was a part of the Technical Services Group, Ericsson Australia, and a Research Engineer at RMIT University. He is currently a Professor of information technology and the Deputy Head of the School of Information Technology, Deakin University, Australia. He leads a team of researchers and Ph.D. students in the broad areas of communication systems and cybersecurity with a focus on emerging domains, such as the IoT, pervasive computing, applied machine learning, and ambient intelligence. His research has been funded by the National Security Science and Technology Branch of the Office of National Security in collaboration with the Defence Signals Directorate, the Australian Research Council, and industry partners. He is the Founding Chair of the Future Network Systems and Security Conference series. He is also an Associate Editor of the journal of Cyber-Physical Systems.