Assessing the Integrity of Traffic Data Through Short Term State Prediction

Doaa Eldowa*, Khalid Elgazzar†, Hossam S. Hassanein*, ‡Tayseer Sharaf, Sumit Shah (CGI USA)
*School of computing, Queen's University, Kingston, ON K7L 3N6, Canada
†ECSE Department, Ontario Tech University, Oshawa, ON L1H 7K4 Canada
‡University of Michigan-Dearborn, Dearborn, USA
eldowa@cs.queensu.ca, khalid.elgazzar@ontariotechu.ca, hossam@cs.queensu.ca, tsharaf@umich.edu, sumit.shah@cgifederal.com

***Abstract*—In this study, we propose an anomaly detection algorithm on sensor traffic data. The algorithm is composed of three distinct steps: temporal detection, spatial detection, and GPS calibration. The temporal detection is based on time series analysis and detects anomalies in real-time when measured sensor values are far offset from expected readings. The spatial detector is used to prune the output of the temporal detector, identifying those anomalies which are not consistent with measurements from neighboring sensors. Both temporal and spatial prediction use the widely adopted ARIMA model. The final step is to compare the predicted speed with the average speed gathered from vehicles equipped with GPS devices and subscribed to provide their data. Experimental results on real data demonstrate that the proposed algorithm effectively differentiates between abnormal traffic events and malicious manipulation of traffic data with an average accuracy of 94%.**

***Keywords*—** Data integrity, ARIMA, Traffic prediction, IoT

I. Introduction

Traffic sensors are widely used all over the world to detect traffic status. This data provides the primary information that affects traffic control system strategies and decisions. Many applications, such as automatic traffic light controller and variable speed limit, adapt the control system according to the measurements of the traffic sensors. Thus, ensuring the quality of the collected data is essential to the reliable operation of the entire system. However, traffic sensors deployed in the open and connected world may be compromised, malfunction, or provide inaccurate measurements due to lifetime issues. So, vital decisions may be made based on false, imprecise, or faulty data [1]. Different factors may affect the quality of traffic sensor data ranging from hardware faults such as detector card broken to communication failures, systematic failures, and electrical failures [2]. In addition to system-level errors, traffic sensors that send their data to the nearest access point are at risk of data manipulation attacks (e.g., a man in the middle attacks). Either traditional traffic sensors such as inductive loop detectors or recently developed Bluetooth-based traffic sensors are highly subject to such kinds of attacks [3].

Although hackers cannot directly control vital traffic control systems such as traffic light controllers, manipulating the data measured and sent by traffic sensors could lead to a more prolonged period of green or red lights that may cause traffic jams or even accidents. Hence, it is fundamental to detect such attacks to avoid these unsafe consequences.

To ensure high quality and integrity of the traffic measurements and detect erroneous data, we propose a comprehensive ARIMA-based machine learning approach to capture normal traffic conditions and compare predictions with real-time measurements to detect any anomalies. Auto-Regressive Integrated Moving Average (ARIMA) [4] is typically applied for temporal-based predictions (i.e., doesn't capture the spatial status of the time series). In this study, we apply ARIMA in both temporal and spatial domains to capture the traffic status progressively over time as well as the correlation with traffic conditions at neighboring segments using sequence analysis. We also calibrate the model with probe data gathered from vehicles equipped GPS devices to improve the detection accuracy. The outcomes serve as an alarm for traffic control systems fed with data from suspicious sensors to take precautionary measures.

II. Related Work

Short-term traffic prediction models can be grouped into parametric and nonparametric models. Nonparametric models include artificial neural networks (ANNs), data mining, and clustering algorithms. Park et al. [5] present a speed prediction algorithm using a neural network model trained with historical traffic data to predict the speed profile in the next 30 minutes. Li [6] uses nonparametric regression to predict traffic flow and leverage MapReduce on the cloud to reduce end-to-end latency to meet real-time requirements. However, the focus was only on temporal analysis at one intersection, ignoring the impact of neighboring intersections.

Parametric models, such as Kalman filter and ARIMA, are based on time series analysis and typically have tuning parameters to adjust the behavior of the model. Kalman filter and its many extensions have been used for short term traffic predictions [7]. Wang and Papageorgio [8] apply an extended version of the Kalman filter to predict traffic flow at specific road segments based on the correlation with neighboring segments. The main constraint they apply though is that road segments should have only either on-ramp or off-ramp, not both. Adaptive Kalman filter [9] has also been used for multi-step ahead traffic flow prediction. The authors propose using the average historical flow as the system's noisy measurements during the prediction interval. Since the traffic volume can vary substantially depending on various external factors, the major drawback in this approach is the high dependency on historical data.

ARIMA models have been widely used for traffic flow prediction due to its superior performance and less processing time [4, 11]. ARIMA also has multiple variations with varying performance and requirements. A comparative study between standard ARIMA and adaptive ARIMA-GARCH has been conducted in [11]. The thesis of their research is that both algorithms provide comparable accuracy, but the processing time of the standard ARIMA is significantly less than that of the adaptive algorithm.

There have also been approaches that use a combination of nonparametric and parametric techniques. For example, Wu et al. [4] propose a hybrid prediction algorithm that uses both clustering of historical data and time series analysis for runtime prediction. The authors classified six clusters from historical data using the K-means clustering algorithm and built an ARIMA model corresponding to each cluster to predict the next flow value in the next hour. Their experimental results show that their hybrid

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

approach performs better than exclusively forecasting based on either historical data or real-time sensor data. The main drawback of their approach is that they did not take spatial correlation into account.

In addition to parametric and nonparametric models, stochastic approaches also have been used for traffic prediction. Qi et al. [10] explored the use of Hidden Markov Models (HMMs) for short-term freeway traffic prediction. They analyzed the sequence of traffic speed observations to build a state transition probability matrix to characterize the transition between different traffic conditions. However, their approach requires long-term time series and stretched transition windows to gather enough information for acceptable prediction accuracy. Raiyn et al. [12] propose a hybrid approach of moving average and exponential smoothing algorithm to detect traffic anomalies. Also, authors in [13] take into account surrounding factors such as weather, location, and the day off of the week with traffic condition when building traffic models from historical data.

Out of all objectives of traffic prediction, anomaly detection stands out due to its paramount importance to traffic control. Anomaly, in this context, is defined as a significant deviation (increase or decrease) in traffic status from expected values. ARIMA modeling has been used for anomaly detection of network segments [14]. The model establishes a profile of the normal behavior of a network segment and then detects any significant variation as abnormal behavior.

All aforementioned detection algorithms consider traffic bursts, such as incidents, as anomalies, which is actually true. However, our approach distinguishes between such legitimate anomalies stem from unexpected traffic events and malicious manipulation of data that present untrue traffic conditions. This paper proposes a robust anomaly detection algorithm on sensor traffic data. The algorithm runs in three complementary phases: temporal detection, spatial detection, and GPS calibration. The temporal detector captures anomalies in real-time sensor values that are significantly offset from historical readings. The spatial detector prunes the output of the time detector, identifying anomalies that are inconsistent with neighboring sensors. The final verification phase is to compare the values with live GPS data collected from vehicles.

Algorithm 1: Anomaly Detection Algorithm
Input: Sensor Valuesflow, speed
Result: True/False

```
Begin
for each time interval T do
    predict next value using last measurements of the same sensor
    predict next value using last measurements of preceding sensor
    if then
        isValid = True;
    else
        mismatch between predicted and measured values
        if GPS probe data avaliable then
            if sensor measurements within the range of GPS data then
                isValid = True;
            else
                isValid = False;
            end
        else
            if mismatch between predicted and measured values occurs at neighbored sensors then
                isValid = True;
            else
                isValid = False;
            end
        end
    end
end
```

III. The Proposed Approach

To ensure the integrity and high quality of the data collected by traffic sensors, we propose a spatiotemporal anomaly detection algorithm. Identifying anomaly is based on the deviation from the normal behavior of the data. Thus, the main idea is to identify temporal anomalies at the first phase, examine whether the measured value at the point of interest correlates with values from neighboring sensors. If it does, then it is a true condition caused by an abnormal event; otherwise, the data is flagged *suspicious*. Further, we can increase the decision confidence level by leveraging data collected from GPS-equipped vehicles sharing their location in real-time, pending data availability at the area of interest and a reasonable penetration rate. While the first phase (temporal phase) uses ARIMA as a time series analysis model to predict the traffic status in the next time interval, the second phase (spatial phase), uses ARIMA as sequence prediction model rather than its typical use as a time series analysis. The phase aims to find the spatial correlation between sensors to distinguish between real abnormal conditions and fake/untrue conditions. Algorithm 1 illustrates this process.

The steps of the proposed algorithm are outlined as follows.

1) Train a temporal ARIMA model on the historical time series to predict the traffic follow in the next time interval.
2) Flag any suspicious sensor that provides real-time measurements that are significantly offset from the predicted value. The offset threshold can be determined experimentally or mathematically from the standard deviation of the dataset over respective intervals to determine reasonable values. Note that it could be unusual traffic conditions caused traffic jam after a significant event or sudden change in the weather.
3) Train a spatial ARIMA model using sequence analysis to predict the correlation between neighboring sensors preceding the traffic follow at the point of interest. This model will determine the number of preceding sensors that have a direct effect on the traffic volume at the current sensor. That is determined by order of p and q of the AR and MA models, respectively. The coefficient of each attribute determines the weight of each of these sensors, on the measurements of the sensor of interest.
4) Calculate the error margin of the spatial model using the average and standard deviation of the training phase.
5) Calculate the absolute difference between the measured value from the sensor and the expected value from the prediction model.
6) If the difference is greater than or equal to the error margin, then the decision is to confirm that this is a suspicious data. The probe data from GPS-equipped vehicles will be used to strengthen or weaken this decision.
7) Use GPS data collected from connected vehicles (given we have a high penetration rate at the area of interest) to confirm the traffic status reported by both the temporal and spatial models. Then, we calculate the average speed and the standard deviation to confirm the decision. We calculate the error margin that allows the valid sensor data to be within the error range with a 99% confidence interval.

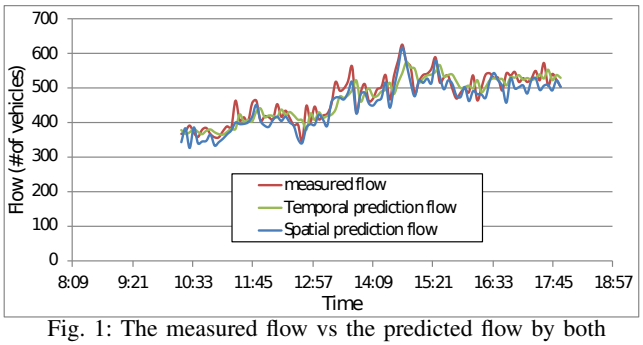

Fig. 1: The measured flow vs the predicted flow by both temporal and spatial predictions.

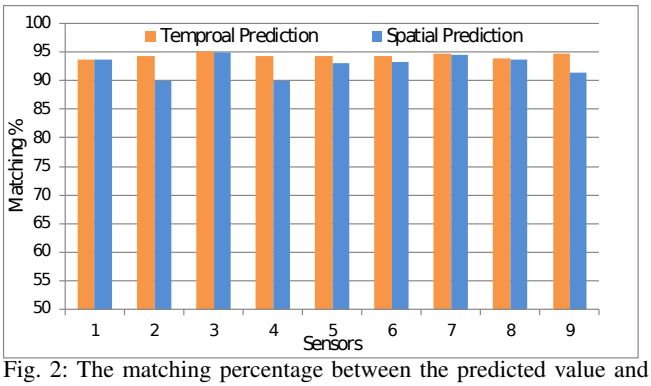

Fig. 2: The matching percentage between the predicted value and the actual measured value.

For the majority of our tested sensors, ARIMA (0, 1, 1) gives the best prediction results for time series data. In the spatial domain, ARIMA (4, 1, 4) gives the best results, which means that a sensor is highly correlated with the four preceding sensor. For GPS calibration, we use the mean speed and standard deviation to set the acceptable speed range. We first check the penetration rate (the proportion of probe vehicles on the road). Yim et al. [15] conclude that 5% is the minimum penetration rate for GPS devices to obtain accurate estimates of the travel time. However, in a recent study they found that only 3% penetration rate provides accuracy comparable with roadside sensors for traffic monitoring. Thus, for all experiments in this study, a minimum of 3% penetration rate is used. Probe data under this level will not be enough for GPS calibration as its values may imprecisely profile the actual traffic status.

IV. Experimental Analysis

Our dataset is collected by the Highway Performance Measurement System (PeMS) which is maintained and operated by the California Department of transportation [16]. The PeMS system collects real-time data from loop detector stations installed at different consequent road segments on the freeway. We train the ARIMA models using the 80-20 data split rule, 80% for training, and 20% for validation. The spatial and temporal models are trained on the same dataset. The temporal model predicts the next value at senor x from its previous time series, while the spatial model uses sequence analysis to predict the next value at sensor x from the values of several preceding sensors in the same traffic direction. Then, the models are used to fit new data to perform prediction and anomaly detection following our proposed algorithm. We use the Mean Absolute Percentage Error (MAPE) to measure the prediction accuracy of ARIMA models. Then, we run two use cases to evaluate how the algorithm differentiates between tampered data and abnormal traffic events.

Figure 1 plots the forecasting results of a 7-hours time horizon of the testing data at one location. We observe that both the temporal and spatial models produce similar results and are very close to the actual measured traffic flow. The figure provides high confidence of the trained models in real-time prediction.

Figure 2 shows the matching percentage $(100 - MAPE)$ between the predicted and actual values for nine sensors (S1-S9). The average matching percentage for all the sensors is 94% with a standard deviation of 1.8. It is noted that the prediction results of the spatial model are very close to the results obtained from traditional time series analysis. This validates our hypothesis that ARIMA is a useful approach not only for time series analysis, but also for spatial sequence analysis.

After validating the prediction accuracy of ARIMA, we now decide on the best aggregation interval of our dataset that yields a high prediction accuracy and provides a reasonable short-term prediction. Typically, higher aggregation intervals provide more transparent patterns of the data, as illustrated in Figure 1. On the other hand, the shorter the prediction interval, the better it is in assessing sensor quality. Thus, we calculate the MAPE for 5, 10, and 15 minutes aggregation intervals. Figure 3 shows a slight improvement for the 15-minutes over the 5-minutes prediction term. However, such a slight improvement does not worth the extra 10 minutes wait to get the prediction results. Therefore, we decided to go with the 5-minutes short-term prediction, which is applied to all the following experiments hereafter.

Besides the validation against measured sensor values, we also conduct a second validation test using GPS probe data with 3% penetration rate to detect traffic anomalies. However, it is essential to ensure that the gathered samples of GPS speeds represent the actual traffic population before the data is valid for verification. In this experiment, we compare the average speed reported by GPS data and the corresponding average speed measured by loop detectors. Table I represents a sample data of speed measurements reported from both GPS devices and loop detectors. For instance, S21 measures a speed of 25.9 mile/hour while at the same time the average speed gathered from 12 vehicles equipped by GPS is 19.9 mile/hour with a 7.7 standard deviation. Thus, for 95% confidence interval, the average GPS speed ranges from 14.8 to 25.1 mile/hour, in which the reported sensor speed (25.9) falls outside the range. However, for a wider 99% confidence interval, the corresponding t-score [1] is 3.1, and the range of speed is 13.1 to 26.8, in which the reported sensor speed falls belong. The available data also shows the at sensor S8 the average GPS speed is 66.5 with a 7.8 standard deviation. Considering the 99% confidence interval ensures that the speed reported by the loop detector is within the GPS speed range.

Figure 4 shows the matching percentage between measured sensor speed and GPS average speed with two different confidence intervals. The 99% confidence interval shows better matching at almost all sensors. Thus, we argue that a high penetration rate is needed to use probe GPS data for anomaly detection validation.

A. Anomaly Detection

The main objective of the proposed algorithm is to raise a flag when data reported by a specific sensor is suspicious while avoiding false alarms when unexpected traffic conditions occur, such as an accident. To evaluate both cases, we study two test case scenarios. In the first scenario, we inject false data to show a tampered sensor reporting traffic congestion. The second use case shows that an actual accident occurs and causes a sharp drop in speed measurements between two sensors. Our algorithm is expected to detect both anomalies, but distinguish between tampered data in the first scenario and actual conditions caused by an abnormal event in the second scenario.

1) Use Case 1: False traffic congestion: The objective of this experiment is to measure the performance of the algorithm in

[1]The t-score is a statistical term used to estimate the population mean from a sampling distribution when the population standard deviation is missing.

TABLE I: Example of loop detector's measured speed and corresponding GPS range of speed.

Sensor ID	Time	LD Speed	Avg. GPS Speed	STDEV GPS Speed	# of GPS Vehicles	Confidence Interval	t-Score	Min. GPS Speed	Max. GPS Speed	LD Speed
S21	11:20	25.9	19.9	7.7	12	0.95	2.2	14.8	25.1	False
S21	11:20	25.9	19.9	7.7	12	0.99	3.1	13.1	26.8	True
S8	10:10	61.6	66.5	7.8	15	0.95	2.1	62.1	70.9	False
S8	10:10	61.6	66.5	7.8	15	0.99	2.9	60.4	72.2	True

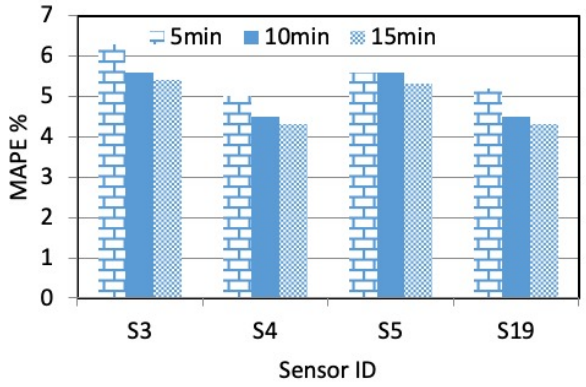

Fig. 3: The MAPE percentage for different predication intervals.

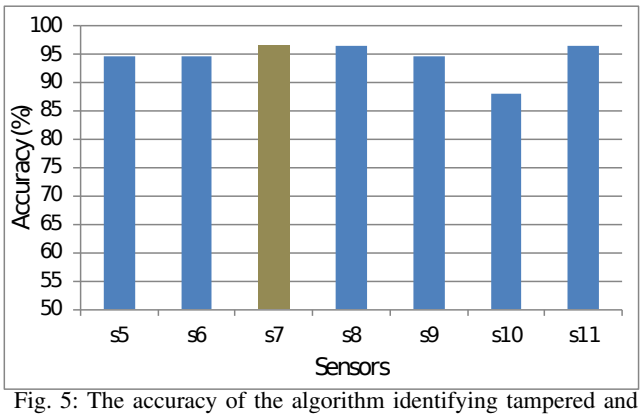

Fig. 5: The accuracy of the algorithm identifying tampered and valid measurements.

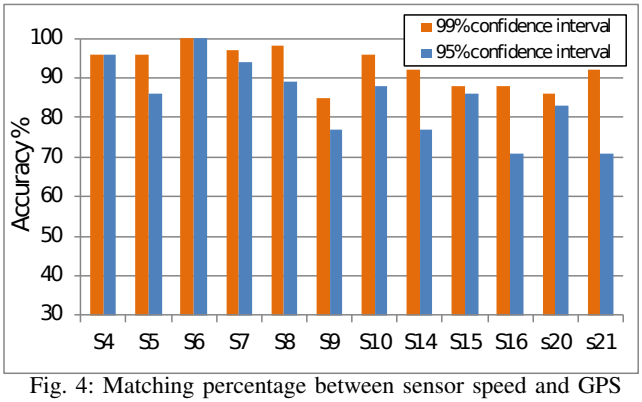

Fig. 4: Matching percentage between sensor speed and GPS average speed with different confidence intervals.

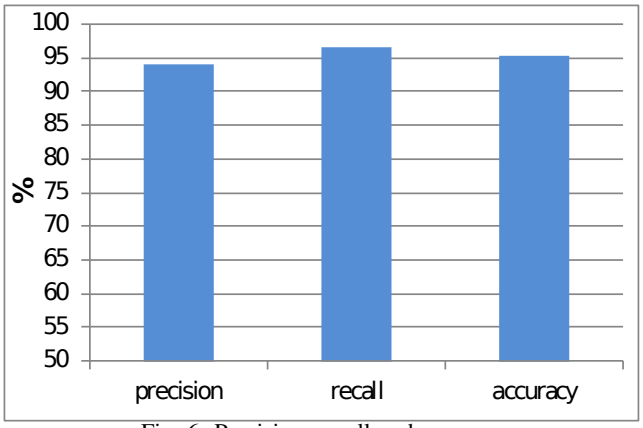

Fig. 6: Precision, recall and accuracy.

assessing the quality of sensor data. We inject erroneous data at sensor 7 and expect a positive alarm. Seven sensors (numbered 5 to 11) are used in this experiment. The average distance between the sensors is 700 meter. For a time interval of 5 hours, data from sensor 7 is tampered and injected with erroneous data that represents traffic jams. The data reported by other sensors are original and untampered. When the temporal model raises alarms about sensor 7, the spatial analysis used for the following sensors (8 to 11) drops the suspicious measurements from sensor 7 and uses sensor 6 instead (i.e., the last known valid sensor).

Figure 5 illustrates the accuracy of the algorithm detecting the correct state of the data at each of the seven sensors, despite S7 is faulty. We also report the *Precision* and *Recall* to measure the sensitivity of the algorithm.

We noticed that the accuracy of the algorithm is consistent with an average of 94% and standard deviation of 3 over the tested sensors, which means that the algorithm can successfully detect untrue conditions reported by tampered sensors (true positive alarms) and normal conditions reported by genuine sensors (true negative) with the same average accuracy. Both precision and recall demonstrate a consistent level of sensitivity when on tampered and untampered sensors, as illustrated in Figure 6. Thus, the proposed approach is unbiased to either positive or negative data.

We also investigate the effect of tampered data on the prediction accuracy of the neighboring sensors of the corrupted S7. Typically, the spatial model at neighboring sensors utilizes the preceding sensor measurements to predict their expected measurements. However, in the case when one of these preceding sensors is flagged as suspicious, the measurements of that sensor will be dropped from the prediction calculation. Thus, we repeat the experiment by dropping the data from S7 and compare the perdition results at the neighboring sensors in both cases. Figure 7 show that the accuracy in both cases is similar to most of the sensors. Hence, we conclude that our algorithm is resistant to tampering or malfunctioning.

2) Use Case 2: Detecting abnormal traffic conditions: The objective of this experiment is to study the performance of the algorithm in distinguishing between incorrect or manipulated data and correct but abnormal conditions from non-recurrent traffic events such as accidents. In this experiment, an accident occurs during the morning hours, which triggers a non-recurrent bottleneck at this time of day. Eleven sensors (s14 to s24) are tested at the time of the accident. The accident occurs between S20 and S21, leading to a sudden slowdown in the reported speed. The effects of the accident quickly build up a traffic jam at neighboring sensors for two hours, as shown in our dataset. We expect that our prediction

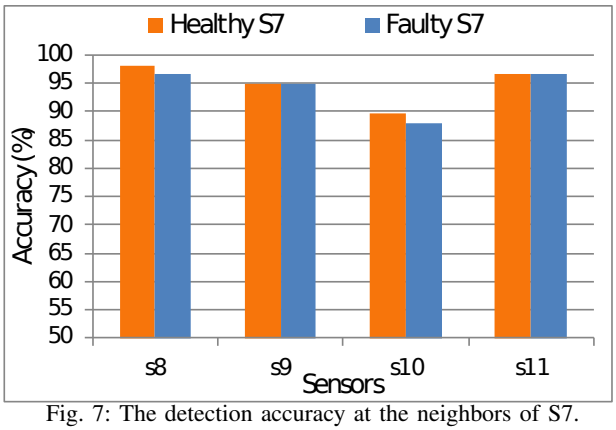

Fig. 7: The detection accuracy at the neighbors of S7.

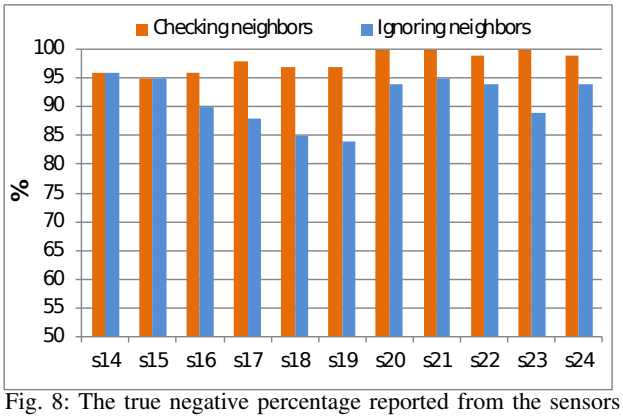

Fig. 8: The true negative percentage reported from the sensors affected by the accident.

algorithm should report no positive alarms as there is no malicious manipulation of sensor data and the sudden traffic jam represents the actual status of the traffic. Thus, this case basically tests the true negative percentage.

Usually, when the algorithm experiences a positive alarm from the temporal prediction model at a specific sensor, it runs the spatial predictor to check the status from the neighbor sensors as well as the temporal conditions reported by these sensors. If the neighbor sensors experience a consistent mismatch between predicted and measured values, the alarm goes negative, indicating an actual status of true conditions.

Figure 8 illustrates the prediction accuracy in this case. The sensors in the middle (19-23) have the highest accuracy as they are the most ones experience a significant speed drop. We also noticed that the propagation of the accident effect impacts the consistency of speed drop between sensors, hence affecting the accuracy at the preceding sensors. The overall average accuracy is 94.6% with a standard deviation of 2.5, which is very close to the accuracy obtained with normal conditions. Hence, we conclude that our algorithm is reliable, resilient to data tampering, and provides high prediction accuracy in both normal and abnormal conditions.

V. Conclusion

In this study, we propose a novel approach that integrates spatiotemporal prediction ARIMA modeling and probe data to detect anomalies in traffic sensor measurements. The algorithm has been evaluated against real-world sensor traffic data provided by Caltrans Performance Measurement System (PeMS). We demonstrate the effectiveness of our algorithm by applying it on two test cases, one contains true data but with a nonrecurring traffic condition such as incidents and the other one contains tampered data. Performance evaluation demonstrates that the proposed approach is able to detect tampered data with high accuracy while maintaining low false positive rates in the case of true data with nonrecurring conditions.

References

[1] Xiao-Yun Lu, ZuWhan Kim, Meng Cao, Pravin Varaiya, Roberto Horowitz, "Deliver a Set of Tools for Resolving Bad Inductive Loops and Correcting Bad Data ", California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 2010.

[2] Chris Bachmann, Matthew J. Roorda, Baher Abdulhai, Behzad Moshiri, "Fusing a Bluetooth Traffic Monitoring System With Loop Detector Data for Improved Freeway Traffic Speed Estimation", *Journal of Intelligent Transportation Systems*, Vol. 12, No. 2, pp. 152-164, 2013.

[3] Guirguis, Mina, and George Atia, "Stuck in Traffic (SiT) Attacks: A Framework for Identifying Stealthy Attacks that Cause Traffic Congestion", The 77 IEEE Vehicular Technology Conference (VTC Spring), 2013.

[4] Cheng-Ju Wu, Thomas Schreiter, and Roberto Horowitz, "Multiple-clustering ARMAX-based Predictor and Its Application to Freeway Traffic Flow Prediction", *2014 IEEE American Control Conference (ACC)*, 2014.

[5] Jungme Park, Dai Li, Yi L. Murphey, Johannes Kristinsson, Ryan McGee, Ming Kuang, Tony Phillips, "Real time Vehicle Speed Prediction using a Neural Network Traffic Model" *The International Joint Conference on Neural Networks (IJCNN)*, 2011, pp. 2991-2996.

[6] Shuangshuang Li, "Implementing Short-term Traffic Flow Forecasting Based on Multipoint WPRA with MapReduce", *IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA)*, 2012.

[7] Jianhua Guo, Wei Huang, and Billy M. Williams, "Adaptive Kalman Filter Approach for Stochastic Short-term Traffic Flow Rate Prediction and Uncertainty Quantification". *Transportation Research Part C: Emerging Technologies*, Vol. 43, pp. 50-64, 2014.

[8] Yibing Wang, Markos Papageorgiou, "Real-time Freeway Traffic State Estimation Based on Extended Kalman Filter: a General Approach", Transportation Research Part B: Methodological, Vol. 39, No. 2, pp. 141-167, 2005.

[9] L. León Ojeda, Alain Y. Kibangou, C. Canudas De Wit, "Adaptive Kalman Filtering for Multi-step Ahead Traffic Flow Prediction" *American Control Conference (ACC)*, 2013.

[10] Yan Qi, Sherif Ishak, "A Hidden Markov Model for Short Term Prediction of Traffic Conditions on Freeways", *Transportation Research Part C: Emerging Technologies* Vol. 43, No. 1, pp. 95-111, 2014.

[11] Chenyi Chen, Jianming Hu, Qiang Meng, Yi Zhang, Short-time Traffic Flow Prediction with ARIMA-GARCH Model", *Intelligent Vehicles Symposium (IV)*, 2011.

[12] Jamal Raiyn, Tomer Toledo, "Real-Time Road Traffic Anomaly Detection", *Journal of Transportation Technologies*, Vol. 4, No. 3, pp. 256-266, 2014.

[13] Hector Gonzalez, Jiawei Han, Yanfeng Ouyang, Sebastian Seith, "Multidimensional Data Mining of Traffic Anomalies on Large-Scale Road Networks", *Transportation Research Record*, Vol. 2215, No. 1, pp. 75–84, 2011.

[14] Eduardo Pena, Sylvio Barbon, Joel Rodrigues, Mario Lemes Proença, "Anomaly Detection Using Digital Signature of Network Segment with Adaptive ARIMA Model and Paraconsistent Logic", *2014 IEEE Symposium on Computers and Communication (ISCC)*, 2014.

[15] Youngbin Yim, "The State of Cellular Probes", California PATH Working Paper UCB-ITS-PRR-2003-25, Institute of Transportation Studies, University of California, Berkeley, CA.

[16] PeMS, California Performance Measurement System, http://pems.eecs.berkeley.edu.