False Data Injection Detection Using Auto Regression Technique

Dianeswarr, Ammar, Tarun, Arslan
MS. Embedded Systems - 2019
UPPSALA UNIVERSITY
Course: Wireless Communication and Networked Embedded Systems

***Abstract*— Sensor motes provide data about different physical processes which is used for information purposes or to drive control systems. False data injection is a significant problem faced in these systems, which affects the overall efficiency of the system and the authenticity of the data collection. The state of the art detection algorithms and using the autoregression technique for this false data injection detection are studied. A test environment in a lab room with a base station in the middle and nine static z1-motes at different locations in the lab room are setup. Detecting false data injection using this technique was successful.**

I. Introduction

Wireless sensor networks are composed of low-energy, small-size, and low-range unattended sensor motes. These motes are used for monitoring and recording information about physical processes or the physical condition of the environment and then transmit this data to a centralised base station for analysis for different application purposes. Since this information can sometimes be used to drive a control system, there is always a chance for false data injection or modification of the actual data by an external attacker so that the control centre makes a wrong decision and that could threaten the safe operation of the whole system. Due to this constant external threat, many false data injection detection techniques exist, but they have limitations like they require an extensive data set for training purposes, need more time to generate an accurate model or affected by environmental conditions. Machine learning is an intelligent method in data mining and is excellent in dealing with data by self-learning. Regression is a technique used for predicting future values.

In this paper, false data injection detection problem by using a machine learning technique known as autoregression to predict the values are addressed. Z1 motes are used for communication, Python libraries to apply the machine learning technique and Dash for the data visualisation.

II. WIRELESS NETWORK SETUP AND DATA EXTRACTION

Initially, a network of 9 motes using the multihop technique is established, where each mote sends a packet to the base station for every 4 seconds. Each packet consists of the temperature information and the respective mote id stored in a structure as elements. All the motes are programmed using Instant Contiki. Every mote is programmed to transmit a random float value to the base station instead of sensing the actual temperature reading to inject false data into the system, for every 60 seconds. When the base station receives a packet, it breaks down the structure and read the incoming packet's mote id. The base station receives data from all the motes and prints all the data on the terminal, which includes actual mote temperature and false temperature readings. Data is extracted continuously from the terminal and put into SQLite Database.

A Challenge encountered with wireless motes: Sometimes the motes become inactive whenever it cannot find a neighbour, and this caused unexpected breaks during the data collection, to counter this, motes are programmed to reset automatically. Whenever they can't find a neighbour, a callback function reboots the watchdog timer which ensures that the mote is reset.

III. MACHINE LEARNING: AUTO REGRESSION TECHNIQUE

Considering the data stored in SQLite DB file, a Python code is used to train an Auto Regression (AR) model on data read from each node. Python library called Pandas. Pandas play a vital role to calculate the AR coefficients and the suitable number of AR lags for each mote depending on the training data. The model (the lags and their correlated coefficients) then is used to generate time sequence of predicted values, and these values are compared with values coming live from sensor motes. With a predefined margin of error (err) for each mote, the system can decide whether the temperature reading is correct or false. After the Auto Regression calculation, both real and predicted values are uploaded back to the SQLite DB file to be used for the next visualisation part. The process described above is repeated every 30 seconds.

IV. LIVE DATA VISUALISATION

To visualise the data live, dash with plotly is used, the data is hosed in the local-host IP address. Figure 1 shows live temperature data from all the motes in a single graph. Data is collected for about 30 minutes. Data includes actual data and false data. False data is randomly generated for every 60 seconds, but mote id 48 gets its changes only after other modes inject false data. For the observation point of view, mote id: 48 is taken. Figure 2 shows the temperature of mote id: 48, three false data are injected at last. Figure 3 shows the predicted and actual value of the mote id 48.

V. CONCLUSIONS

The use of an auto-regression technique to detect false data injection is an easy method with fairly good results. In future, the efficiency of the model can be improved by increasing the accuracy of the predicted value by using more advanced techniques like Gaussian regression which work better with small data sets as well.

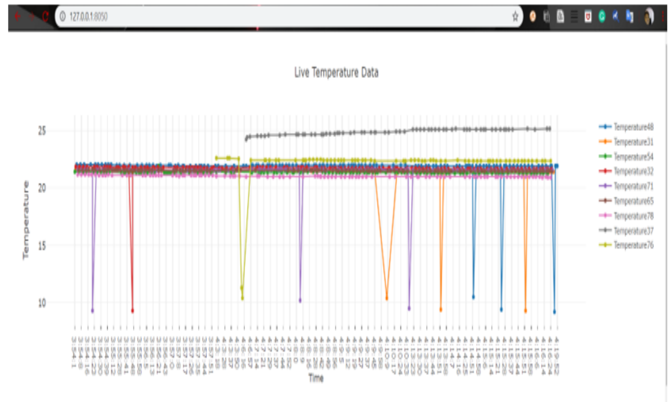

Fig. 1. Temperature vs Time of 9 motes

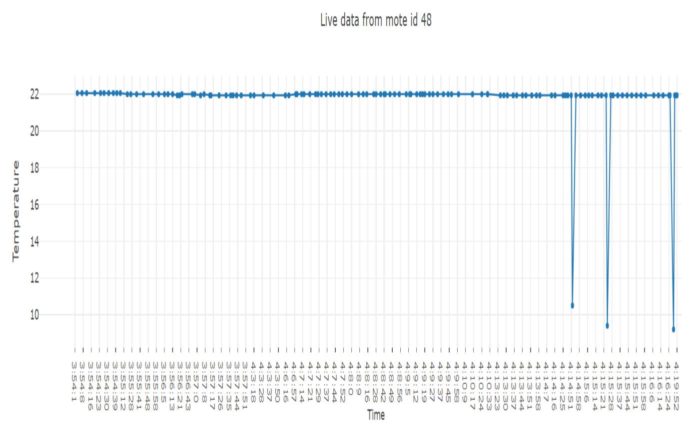

Fig. 2. Temperature data of Mote 48

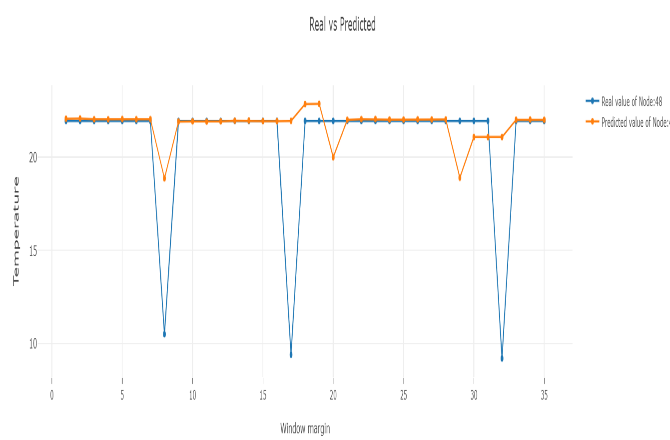

Fig. 3. Real vs Predicted values of Mote 48

REFERENCES

[1] https://github.com/contiki-os/contiki/blob/master/examples/rime/example-multihop.c

[2] https://pandas.pydata.org/
[3] https://otexts.com/fpp2/AR.html
[4] https://dash.plot.ly/live-updates
[5] https://pythonprogramming.net/live-graphs-data-visualization-application-dash-python-tutorial/
[6] https://makersportal.com/blog/2018/2/25/python-datalogger-reading-the-serial-output-from-arduino-to-analyze-data-using-
[7] https://pythonprogramming.net/sql-database-python-part-1-inserting-database/
[8] https://pythonprogramming.net/sqlite-part-3-reading-database-python/
[9] https://pythonprogramming.net/graphing-from-sqlite-database/