Data Mining (CS F415) Term Project

Anomaly and Misbehaviour detection in VANETs

(Phase 2)

Report by: -

Ruchir Kumbhare (2019B5A70650P)

Shaurya Marwah (2019B3A70459P)

Introduction

Vehicular Ad-hoc Networks (VANETs) are wireless networks that allow vehicles on roads to communicate with each other and with roadside infrastructure to achieve the goal of an Intelligent Transportation System (ITS). VANETs are designed to improve road safety, traffic efficiency, and passenger comfort by enabling vehicles to share information about road conditions, traffic congestion, and potential hazards.

Importance of the task at hand

Since VANETs are wireless networks, they are susceptible to various forms of network attacks and faults. As the safety and reliability of these networks are of utmost importance, multiple proposals have been made regarding identifying incorrect and malicious information being shared across the wireless network.

Prior work

The paper by Alladi et al. [1] proposes a machine learning-based approach for anomaly detection in VANETs. Their approach uses a sequence reconstructor unit to reconstruct normal vehicular messages and detect anomalies based on the difference between the original and reconstructed messages.

Another paper by Alladi et al. [2] proposes a deep learning-based misbehaviour classification scheme for addressing security issues.

Dataset used

The dataset used is Veremi Extension [3], [4], developed by Joseph Kamel, is a misbehaviour detection dataset created for Cooperative Intelligent Transport Systems (C-ITS). It is used to compare and reproduce different results in misbehaviour detection studies. The dataset consists of peer-to-peer messages sent on VANET and includes a set of attacks and a ground truth file.

The data fields present in the dataset are the send time, sender ID, sender pseudo ID, message ID, and position-based information - position, speed, acceleration, and heading. We used only X, Y coordinates of position and speed among the other data fields for the creation of our customized dataset.

The Dataset contains 19 types of anomalies, these can broadly be divided into two types, Fault and Attack. The dataframes generated in all the Python Notebooks have been made to reflect these broad anomaly types. Each entry of the dataframe has been given one of three labels:

0 for Genuine behavior

1 for Anomalies caused due to some fault

2 for Anomalies caused to an attack

DETAILS OF ANOMALY AND GENUINE TYPES CONSIDERED

Type	Class	Description
0	Genuine	Genuine behavior
1	Anomaly (Fault)	Constant position
2	Anomaly (Fault)	Constant position offset
3	Anomaly (Fault)	Random position
4	Anomaly (Fault)	Random position offset
5	Anomaly (Fault)	Constant speed
6	Anomaly (Fault)	Constant speed offset
7	Anomaly (Fault)	Random speed
8	Anomaly (Fault)	Random speed offset
9	Anomaly (Fault)	Delayed messages
10	Anomaly (Attack)	Disruptive
11	Anomaly (Attack)	Data replay
12	Anomaly (Attack)	Eventual stop
13	Anomaly (Attack)	DoS
14	Anomaly (Attack)	DoS random
15	Anomaly (Attack)	DoS disruptive
16	Anomaly (Attack)	Data replay sybil
17	Anomaly (Attack)	Traffic congestion sybil
18	Anomaly (Attack)	DoS random sybil
19	Anomaly (Attack)	DoS disruptive sybil

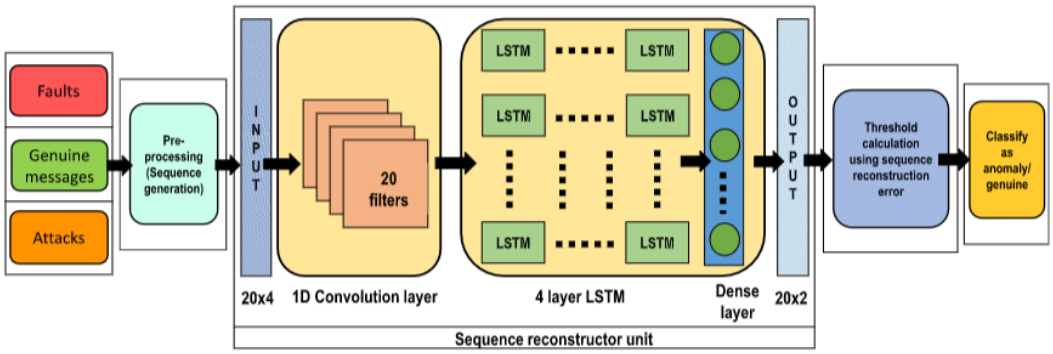
The dataframes thus obtained are as follows (after the data has been standardised appropriately),

	label	posX	posY	spdX	spdY
0	0	-0.733647	-2.038607	0.064290	0.123897
1	0	-0.733858	-2.031121	0.037468	0.374456
2	0	-0.735468	-2.016168	0.012002	0.612341
3	0	-0.737286	-1.993106	-0.015018	0.864757
4	0	-0.740588	-1.963119	-0.067851	1.093883
...	...	...	...	...	...
10737	0	-0.560248	0.469826	1.037734	0.848426
10738	0	-0.527329	0.467900	1.519191	-0.485500
10739	0	-0.503352	0.431303	0.872424	-1.097320
10740	0	-0.487958	0.402649	0.574459	-0.675031
10741	0	-0.480615	0.387312	0.275578	-0.251382

Our attempts at recreating the results of publish papers

DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs [1]

The paper proposes a deep learning-based approach for anomaly detection in VANETs. The architecture proposed implements a preprocessing of data into chunks of 20 consecutive entries. These are then passed into a sequence reconstruction unit, which takes in (20,4) inputs and gives a (20,2) output, which is then passed through the thresholding algorithm to predict if the input has an anomaly.



Our implementation of the same is present in the .ipynb file named as '**Sequence_Reconstruction_&_Thresholding_Algorithm'**.

The attempt made by us to recreate the model as mentioned in the paper provides us inconclusive results since the paper is quite ambiguous in the usage of the results obtained by the thresholding algorithm to update the weights of the Sequence reconstructor unit.

A Deep Learning Based Misbehavior Classification Scheme For Intrusion Detection In Cooperative Intelligent Transportation Systems [2]

This paper proposes a deep learning-based misbehavior classification scheme for various anomalies present in the Veremi Dataset. The classification is into three categories, Genuine, Anomaly (Fault) and Anomaly (Attack), as described previously in the Dataset section. All the implementations in this paper include LSTMs. We have attempted to implement the most basic model in the paper 3-LSTM, which has three LSTM layers in the .ipynb files named as **'LSTM_classification'** and **'LSTM_classification_2'**.

The attempt made by us to recreate the model as mentioned in the paper provides us inconclusive results since the paper does not mention how the final predicted labels are obtained from the LSTM model. In our attempt, the LSTM model gives a value between 0 & 1, for which there isn't a definite method to assign a label logically. This is a major flaw in the above approach.

Vanilla Neural Network based approach

On not obtaining any conclusive results by implementing [1], [2], we implemented a straight forward neural network approach which has multiple linear layers, coupled with ReLU activations. This model attempts classification into three categories, Genuine, Anomaly (Fault) and Anomaly (Attack), as described previously in the Dataset section. The implementation is present in the .ipynb file named as **'Straightforward_neural_network'**. The results obtained on the test data were as follows

	precision	recall	f1-score	support
0	0.72	0.65	0.68	270
1	0.60	0.63	0.61	103
2	0.60	0.67	0.64	166
accuracy			0.65	539
macro avg	0.64	0.65	0.64	539
weighted avg	0.66	0.65	0.65	539

The accuracy obtained on the test data is found to be ~65% which is not a good result in our opinion. The features considered by us, Position and Speed only (these were selected by referring to [1]), by intuition do not contain sufficient information to encode the misbehaviour information in VANETs.

K Nearest Neighbour Approach

This is a more straight forward approach than any neural network as attempted in the previous section. This model attempts classification into three categories, Genuine, Anomaly (Fault) and Anomaly (Attack), as described previously in the Dataset section.

The implementation is present in the .ipynb file named as **'KNN'**. The results obtained on the test data were as follows

	precision	recall	f1-score	support
0	0.78	0.73	0.76	432
1	0.70	0.73	0.71	230
2	0.72	0.77	0.74	228
accuracy			0.74	890
macro avg	0.73	0.74	0.74	890
weighted avg	0.74	0.74	0.74	890

The accuracy obtained on the test data is found to be ~74% which although not great is a better accuracy compared to the previous neural network based approach.

Conclusion

The currently published papers by Alladi et al. [1], [2], in the domain of anomaly detection/classification in VANETs do not have any publicly available codes, nor any justifications for the approaches taken which makes reproducing their results nearly impossible.

The straight forward approaches taken by us in the end do not give satisfactory results. They require more extensive research in fine tuning the deep learning models, which requires more expertise than what we possess right now.

References

1. T. Alladi, B. Gera, A. Agrawal, V. Chamola and F. R. Yu, "DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs," in IEEE Transactions on Vehicular Technology, vol. 70, no. 11, pp. 12013-12023, Nov.2021, doi: 10.1109/TVT.2021.3113807.

2. Tejasvi Alladi, Varun Kohli, Vinay Chamola, and F. Richard Yu. 2022. "A Deep Learning Based Misbehavior Classification Scheme For Intrusion Detection In Cooperative Intelligent Transportation Systems". Digital Communications And Networks. doi:10.1016/j.dcan.2022.06.018.

3. J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi extension: A dataset for comparable evaluation of misbehavior detection in vanets," in Proc. IEEE Int. Conf. Commun., 2020, pp. 1–6.

4. "VeReMi Extension," 2020, Accessed: March. 2023. [Online]. Available: https://github.com/josephkamel/VeReMi-Dataset.