CS581 Project outline – Exploring TCNs for Network Anomaly Detection.

By Sanket Mehrotra and Kaustubh Jawanjal

Introduction

A. Motivation and Proposal

Over the past decade, RNNs and LSTMs have become the default starting point for sequence problems. However more recently Temporal Convolutional Networks (TCNs) are being hailed across the board as the successors to RNNs. We seek to try to show this ourselves in an embedded context for anomaly detection in small LANs used as in-vehicle communication networks also known as CANs. TCNs are not part of a standard keras/tensorflow API, so we will be working with a reference implementation based on the code provided by the authors of [ref 6]. For comparison, we implement LSTMs via keras models. We propose to implement both of these architectures with reference to anomaly detection in networks, a task that requires learning short-term/long-term dependencies and compare their performance. We also plan to compare optimized TCNs as a third model in this project.

Definitions

From our lectures and after going through a few papers, we came across an understanding of certain standard properties of time-series data. Some of these do not apply in our dataset but are used for various analyses across the field of network anomaly detection and anomaly detection in general. We briefly define these and try to illustrate them with examples.

1. Time Series: Sequence data, also known as time-series data, is a set of time-ordered records. A time series can be univariate (d=1) or multivariate (d>1). A univariate time series has one time-dependent attribute. A multivariate time series is used to simultaneously capture the dynamic nature of multiple attributes.[5]
2. Time series properties [12]:
    a. Trend: Trend is defined as the general tendency of a time series to increment, decrement, or stabilize over time.
    b. Seasonality: Seasonality is defined as the existence of repeating cycles in a time series.
    c. Stationarity: A time series is stationary (non-seasonal) if all its statistical features, such as mean, and variance are constant over time.
3. Anomaly detection approaches for time-series data: Fig 1 contains some commonly used approaches in anomaly detection in Time Series, we plan to implement the LSTM and the TCN. Other comparable approaches are HTMs, LSTM Auto-encoders and domain specific - stochastic measures [3].

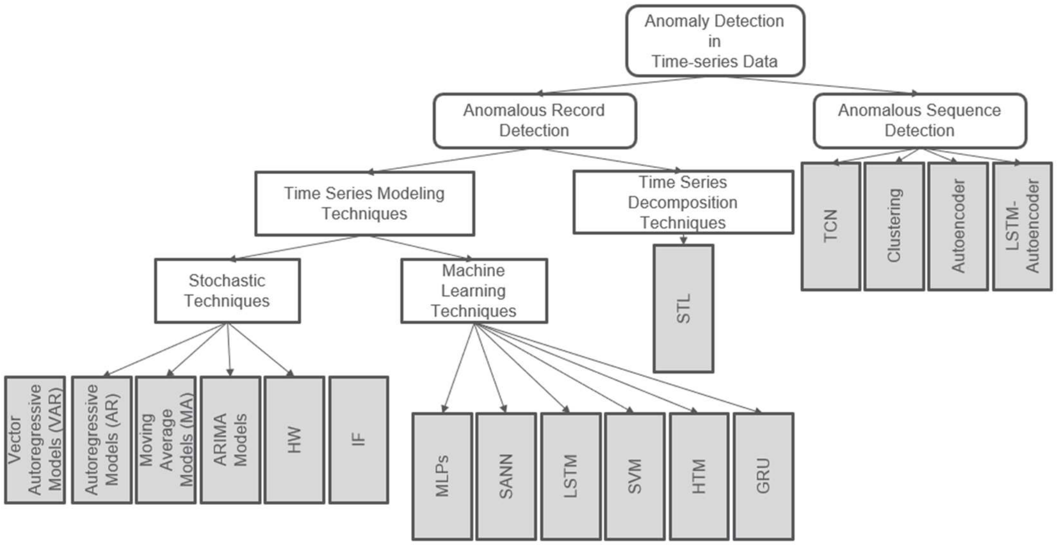

Anomaly Detection in Time-series Data
Anomalous Record Detection
Anomalous Sequence Detection
Time Series Modeling Techniques
Time Series Decomposition Techniques
TCN
Clustering
Autoencoder
LSTM-Autoencoder
Stochastic Techniques
Machine Learning Techniques
STL
Vector Autoregressive Models (VAR)
Autoregressive Models (AR)
Moving Average Models (MA)
ARIMA Models
HW
IF
MLPs
SANN
LSTM
SVM
HTM
GRU

Figure 1: A taxonomy of Anomaly Detection techniques

4. Anomalies and Outliers: As we discussed in class, anomalies are a subset of outliers, which also contain another category which can be defined as noise. In our experiments, we plan to combine these two categories under the umbrella term “outliers”.

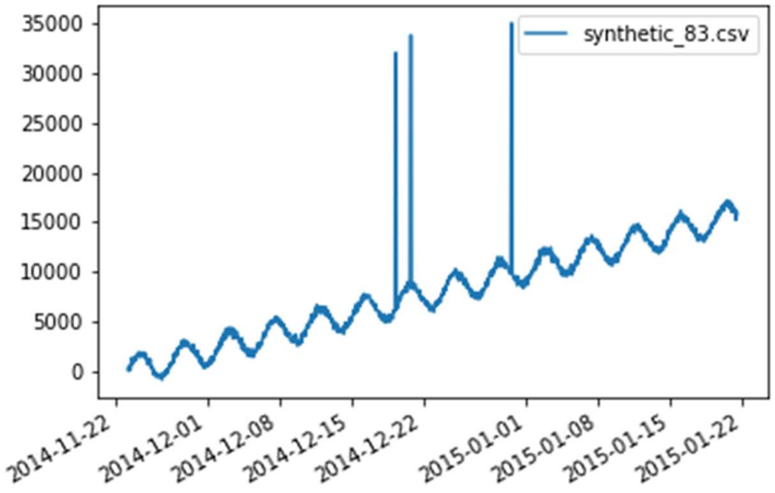


Figure 2: A time series displaying a regular seasonality and increasing trend. Outlier points are also visible.

5. Controller Area Networks: We started by getting familiar with CAN networks and how they work, since neither of us have never worked with CANs before. A CAN is a standardized serial communication protocol that provides efficient error detection mechanism for stable transmission and fast recovery [3]. It is the most commonly used broadcast-based communication protocol in Automotive systems. There are 4 types of standard data frames used in the CAN protocol (1) Data frames, (2) Remote frames, (3) Error frames, (4) Overload frame. The broadcast nature of the CAN protocol make it such that there is no way to check the origin of a message. Herein lies the key vulnerability that can be exploited by malicious attackers.

Dataset

1. CAN Intrusion Dataset - https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

A dataset prepared by the authors of ref #3, in which they test a non-DNN based approach to detect network intrusion detection in CAN networks. I've emailed them for access to their dataset, they are yet to reply.

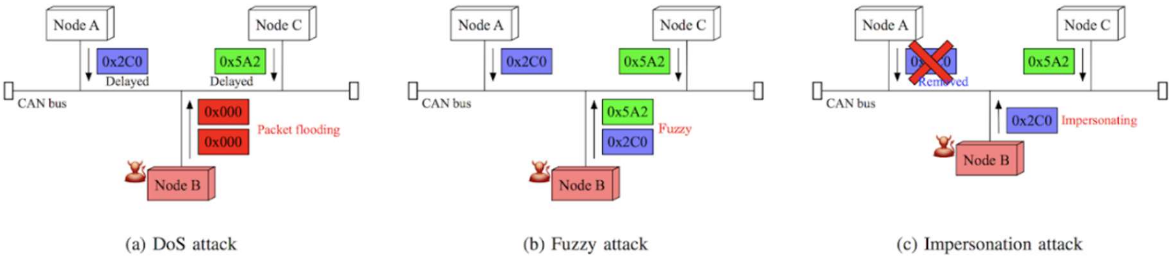
Size: 4 files totaling up to ~ 400 MB

On their site, they describe the dataset as follows:

This dataset is a collection of 4.4 million CAN attack related messages divided according to a specific type of attack that they simulate. The attacks covered are:

1. CAN Denial of Service attacks: Attacks in which a 0x000 can ID message is injected into the network in short cycles.
2. Fuzzy attack: Injecting messages of spoofed random CAN ID and data values.
3. Impersonation attack: Injecting messages impersonating nodes

These attack messages are mixed with regular state CAN messages that are normally exchanged in a network.



(a) DoS attack

(b) Fuzzy attack

(c) Impersonation attack

Figure 3: Types of CAN attack data Image Ref: https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

Attack type	# of messages
DoS Attack	656,579
Fuzzy Attack	591,990
Impersonation Attack	995,472
Attack free state	2,369,868

Data Preprocessing and Analysis [with ref to INDRA and OTIDS papers]

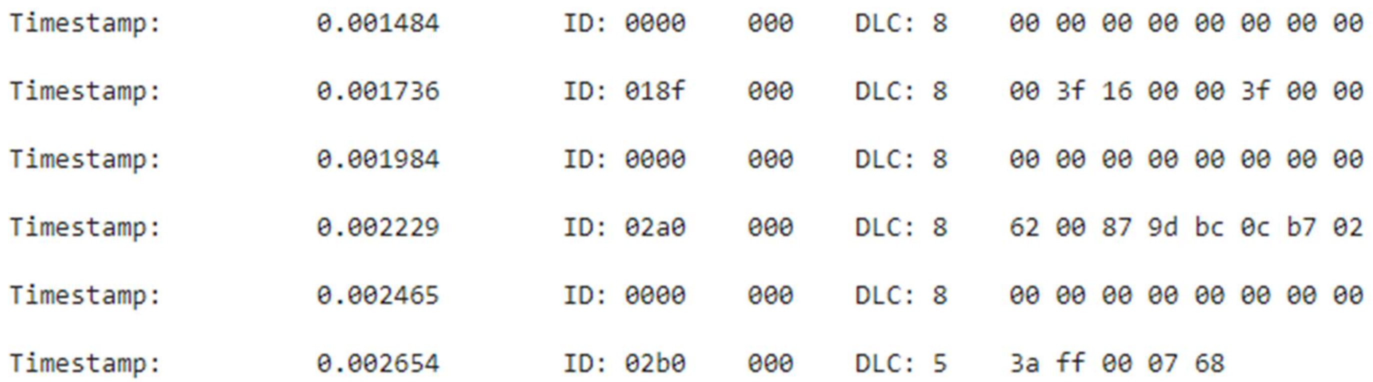
Formatting -> Transformation

Cleaning -> choosing proper data types for analysis

Graphs -> all four types of datasets

Things to be careful of – Class Imbalance Bias, Stdzn, Normalzn, design of data [ref 12]

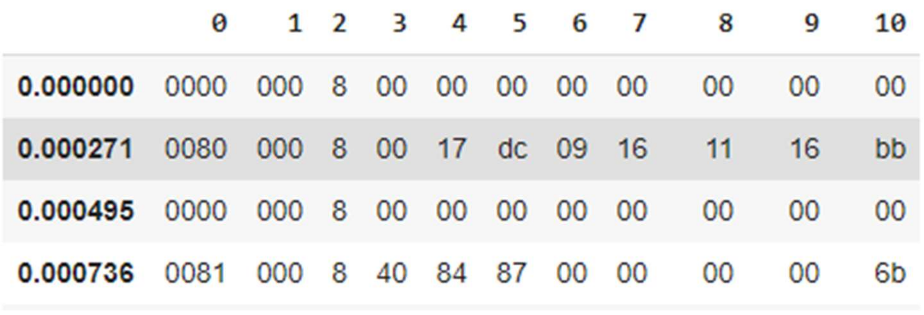
The OTIDS dataset was provided in a format not directly processable by a ML model. Formatting, exploration and certain data preprocessing decisions were to be made in order to prepare our data to be provided as input to the LSTM and TCN models we were preparing. The first step in the data processing pipeline was to convert a fixed delimited file in the format shown below, into a pandas dataframe, which after processing, was written into a csv acceptable as input to a ML model.



```
Timestamp:          0.001484        ID: 0000    000     DLC: 8    00 00 00 00 00 00 00 00
Timestamp:          0.001736        ID: 018f    000     DLC: 8    00 3f 16 00 00 3f 00 00
Timestamp:          0.001984        ID: 0000    000     DLC: 8    00 00 00 00 00 00 00 00
Timestamp:          0.002229        ID: 02a0    000     DLC: 8    62 00 87 9d bc 0c b7 02
Timestamp:          0.002465        ID: 0000    000     DLC: 8    00 00 00 00 00 00 00 00
Timestamp:          0.002654        ID: 02b0    000     DLC: 5    3a ff 00 07 68
```

Figure 4: Initial Dataset format: Fixed Width

When considering the types of the data to convert to, we chose to split the data into single precision floats everywhere except the data column, where we kept the data split into 1-byte columns.



	0	1	2	3	4	5	6	7	8	9	10
0.000000	0000	000	8	00	00	00	00	00	00	00	00
0.000271	0080	000	8	00	17	dc	09	16	11	16	bb
0.000495	0000	000	8	00	00	00	00	00	00	00	00
0.000736	0081	000	8	40	84	87	00	00	00	00	6b

Figure 5: Processed Data

We then checked for possible values in each column and replaced NaNs with ‘00’ byte values. This pipeline has been applied to all four attack datasets and the Normal-state dataset.

Reference Papers and Related work:

1. Anomaly Detection of CAN Bus Messages Using a Deep Neural Network for Autonomous Vehicles by Aiguo Zhou ,Zhenyu Li ,and Yong Shen - https://doi.org/10.3390/app9153174 - DNNs in CAN network detection
2. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling by Shaojie Bai, J. Zico Kolter, and Vladlen Koltun - (https://arxiv.org/pdf/1803.01271.pdf) - TCNs being compared to RNNs for simple tasks
3. OTIDS: A Novel Intrusion Detection System for In-vehicle Network by using Remote Frame by Hyunsung Lee, Seong Hoon Jeong and Huy Kang Kim, PST (Privacy, Security and Trust) 2017 (https://www.ucalgary.ca/pst2017/files/pst2017/paper-67.pdf) - CAN Dataset
4. https://www.researchgate.net/publication/322814361_A_Distributed_Anomaly_Detection_System_for_In-Vehicle_Network_using_HTM - New NN technique for similar application of in-vehicle network detection.
5. http://urban-sustain.org/papers/GhoshIEEEBigData.pdf - An application of Autoencoders LSTMs for anomalous records and sequence detection.
6. https://github.com/locuslab/TCN,
7. https://github.com/philipperemy/keras-tcn#keras-tcn
8. A framework for end-to-end deep learning-based anomaly detection in transportation networks. NeemaDavis,GauravRaina,KrishnaJagannathan https://www.sciencedirect.com/science/article/pii/S2590198220300233
9. Palisade: A framework for anomaly detection in embedded systems. Sean Kauffman, Murray Dunne ,Giovani Graciolib, Waleed Khan, Nirmal Benann, Sebastian Fischmeister https://www.sciencedirect.com/science/article/pii/S1383762120301545
10. Dupont, Guillaume; Lekidis, Alexios; den Hartog, J. (Jerry); Etalle, S. (Sandro) (2019): Automotive Controller Area Network (CAN) Bus Intrusion Dataset v2. 4TU.ResearchData. Dataset. https://doi.org/10.4121/uuid:b74b4928-c377-4585-9432-2004dfa20a5d – an Alternate dataset
11. https://developers.google.com/machine-learning/problem-framing/formulate - Formulating your problem as an ML problem.
12. R. Adhikari and R. K. Agrawal, An Introductory Study on Time Series Modeling and Forecasting. LAP LAMBERT Academic Publishing, 2013
13. https://github.com/etas/SynCAN - A third popular dataset commonly used to benchmark intrusion detection methods.
14. INDRA: Intrusion Detection using Recurrent Autoencoders in Automotive Embedded Systems by Vipin Kumar Kukkala, Sooryaa Vignesh, Sudeep Pasricha