Anomália detekció gépjárművek CAN forgalmában

Verbőczy Kristóf

2017. szeptember 9.

1. Feladat bemutatása

Manapság a gépjárművek számos elektronikus vezérlőegységgel rendelkeznek, amelyek között a kommunikáció a CAN buszon zajlik. Ha egy támadó sikeresen manipulálja a CAN buszon áthaladó adatforgalmat, főleg üzenetek beszúrásával, akkor átveheti a jármű felett az irányítást. Ennek elkerülésére szolgál az anomália detekció. Lényege, hogy valamilyen módon nagy bizonyossággal meg lehessen különböztetni az eredeti adatokat a támadó által beszúrtaktól.

Ahhoz, hogy meg lehessen különböztetni az eredeti és a támadó által beszúrt adatokat, szükségesek a múltbeli adatok, tehát log-olni kell a teljes CAN forgalmat. Ezt a funkcionalitást valósítja meg a CAN logger. A kutatómunka elején rendelkezésre állt egy Python implementációjú CAN logger, azonban ennek a teljesítménye nem volt teljes mértékben kielégítő. Az első feladat az imént említett programmal funkcionálisan teljesen megegyező, de gyorsabb C implementációjú CAN logger elkészítése volt.

A feladat a továbbiakban az anomália detekció megvalósításából, és a különböző paramétereinek teszteléséből állt. Mint az később látható lesz, viszonylag sok paraméterrel rendelkezik az anomália detekció, így ezeknek a kipróbálása és értékelése komplex feladatnak bizonyult.

2. CAN logger

A feladat kihívása a Python implementációval teljesen megegyező funkcionalitású C program elkészítése. Ennek egyik nehézsége, hogy Python esetében rendelkezésre áll a *can* csomag, amely rendelkezik a log-oláshoz szükséges metódusokkal, viszont ennek a megvalósítása el van fedve a felhasználó elől. A másik nehézség, hogy C esetében a GPIO használata komplikáltabb, mint Python esetében, így ennek a megismerése is időt igényel.

Az időigényesebb megvalósításért cserében a C implementáció sokkal gyorsabb lett, mint a Python implementáció. Ezeknek a tesztelése már felvett CAN üzenetek visszajátszásával történt, különböző paraméterek mellett. A 4 adatbájttal végzett mérés eredményét mutatja az alábbi két táblázat.

1	250k	500k	1M
10	1	1	0,996633333
30	1	1	0,986233333
50	1	1	0,755
70	1	0,998533431	0,582733333
90	1	0,939637358	0,456933333

1. táblázat. Python kód

1	250k	500k	1M
10	1	1	1
30	1	1	1
50	1	1	1
70	1	1	1
90	1	1	1

2. táblázat. C kód

3. Anomália detekció

Az anomália detekció lényege, hogy jelezni lehessen, ha olyan adat érkezik, amely jelentősen eltér az eddigi adatoktól. Például ha egy autó 90 kilométer/óra sebességgel közlekedik, akkor a 89 km/h elfogadhatónak tűnik, viszont a 280 km/h már gyanús, hogy nem jó, ekkor már jelezni kell, hogy valamiféle hiba történt.

Ahhoz, hogy meg lehessen állapítani egy adatról, hogy mennyire tér el az őt megelőzőektől nyilvánvalóan szükség van a múltbeli adatok tárolására. Továbbá egy módszerre, algoritmusra, amely képes megállapítani, hogy az adott eltérés jelentős-e vagy sem. Erre a célra az **ARIMA** került felhasználásra.

3.1. ARIMA

Az **ARIMA** kifejtve *Autoregressive Integrated Moving Average (autoregresszív integrált mozgóátlag)*, képes idősorok jövőbeli értékét előrejelezni a múltbeli adatok alapján. Három paraméterrel szokás jellemezni egy **ARIMA** modellt, ez a *(p, d, q)* hármas. A *p* azt adja meg, hogy az előrejelzéshez az utolsó hány értéket használja fel, a *q* pedig azt, hogy az utolsó hány hibáját az előrejelzéseknek. A *d* a differenciálás foka, ennek a jelentése, hogy nem a megadott adatokon dolgozik a modell, hanem azok különbségein.

Működését tekintve szüksége van egy tanító halmazra, ami jelen esetben egy korábbi vezetés alkalmával rögzített CAN forgalom. Ez alapján egy modellt épít fel, ami tulajdonképpen néhány *(p + q darab)* együtthatót takar. Ezen együtthatók alapján képes a megfelelő számú múltbeli adatból egy előrejelzést adni a következő értékre.

A kutatás során rendelkezésre állt több vezetésből származó CAN forgalom, de a különböző adatok közül csak a fordulatszám adatokat sikerült dekódolni, ezért minden tesztelés ezeken zajlott.

Mivel az **ARIMA** modellek három értékkel paraméterezhetők, szükséges volt kipróbálni, hogy a fordulatszám adatokat melyik paraméterek követik a legjobban. Körülbelül 15 paraméterhármas került kipróbálásra, az eredmények azt mutatták, hogy a *(2,1,1)* és a *(10,1,1)* közelítik a legjobban a tényleges adatokat. Ennek megállapítása a maximális és az átlagos eltérések alapján történt.

Elképzelhető, hogy több **ARIMA** modell kerül használatra a jövőbeli értékek előrejelzésére, olyan értelemben, hogy külön tanító halmaz alapján készülnek a modellek. Például három részre osztva, ahol az egyik a gyorsulás, másik az egyenletes haladás, a harmadik a lassulás fázis. Ilyenkor az előrejelzésnél az a cél, hogy mindig a megfelelő modell alapján történjen az érték előrejelzése. Ígéretesen hangzott ez az ötlet, de a tapasztalat azt mutatta, hogy nem ad

jobb eredményt, mint az egy modelles megoldás, sőt annál egy kicsit rosszabb az eredménye. Ennek lehetséges oka, hogy nehéz megtalálni azt, hogy éppen melyik modell alapján kell az előrejelzést végezni.

3.2. Anomália detekció ARIMA felhasználásával

Első megközelítésben az volt az ötlet, hogy az **ARIMA** modell predikcióját összehasonlítva a tényleges értékkel megmondható az adott értékről, hogy hibás-e vagy sem. Viszont a mérési eredmények azt mutatták, hogy a legjobb paraméter esetében is 108 volt az eltérés az előrejelzett és a tényleges érték között, ami a fordulatszámértékeknél *(800-4000)* egészen nagy eltérésnek mondható. Ezért ez a módszer ilyen formában nem alkalmazható anomália detekcióra.

Egy másik módszert kell használni. Ez a módszer egy olyan támadómodellre alkalmas, ahol feltételezzük, hogy a hamis adatok nagyobb frekvenciával érkeznek, mint az eredeti adatok. Ennek a határairól később lesz szó.

Ennek a módszernek az alapja a már említett **ARIMA** előrejelzés, a predikált értéke összehasonlítása a tényleges értékkel. Viszont ez esetben nem az számít anomáliának, ha egyszer a határon kívül van a becsült érték; itt ezt *figyelmeztetésnek* hívjuk. Hanem az, ha egy adott időablakon belül meghatározott számú figyelmeztetés érkezik. Ennek az előnye, hogy nem fog fals pozitív eredményeket adni, mert nagy eltérések nem követik egymást a normál működés során, ezért ha több ilyen van, akkor feltételezhető a támadás.

Látható, hogy a módszer esetében több paraméter is szerepel. Ezek az alábbiak:

- *hamis adatok frekvenciája:* azt adja meg, hogy egy másodperc alatt hány hamis adat került beszúrásra, az eredeti fordulatszám adatok frekvenciája 100 Hz;
- *időablak:* azt mutatja meg, hogy mekkora az időintervallum mérete, amiben a figyelmeztetések száma kerül vizsgálatra;
- *küszöbérték:* azt mutatja meg, hogy mekkora eltérése kell, hogy legyen egy értéknek a becsülttől, hogy figyelmeztetésként kerüljön értelmezésre;
- *figyelmeztetések száma:* azt mutatja meg, hogy az adott időablakon belül hány figyelmeztetés után mondható, hogy támadás van a rendszerben.

3.2.1. Eredmények

- *Frekvencia:* tesztelésre került az eredeti frekvenciához képest 1-szeres, 2-szeres, 5-szörös és 10-szeres frekvencia. Az 1- és 2-szeres esetében nem voltak olyan paraméterek, amelyek megfelelő eredményt adtak volna. 5-szörös esetében már voltak, 10-szeres esetében még több volt.
- *Időablak:* 100 ms, 500 ms és 1000 ms került tesztelésre. A 100 ms sehol sem került be a megfelelő eredményt adó paraméterek közé. Az 500 ms és 1000 ms közel azonos eredményt adott.
- *Küszöbérték:* a tesztelt küszöbértékek: 3, 5, 7, 10. Érdekes módon nem volt jelentős különbség közöttük.

- *Figyelmeztetések száma:* 85, 95 és 100 voltak a tesztelt értékek. A *(2,1,1)* **ARIMA** modell esetében a 100 tűnt a legjobbnak, a *(10,1,1)* **ARIMA** modell esetében pedig a 95.

Az eredmények arra engednek következtetni, hogy ez a módszer a gyakorlatban is használható megoldás lehet.

A tesztelés kizárólag a fordulatszám adatok alapján történt, a gépjárművek sok más adatot is küldenek a CAN buszon. Elképzelhető, hogy az egyes típusokra különböző paraméterek adnak kielégítő eredményt. Ha esetleg a gépjárműgyártásban alkalmaznak ilyesmi technikát, akkor érdemes kipróbálni, hogy az egyes típusokra mely paraméterek adják a legjobb eredményt.