

Gépi tanulás alkalmazási lehetőségei hibadetektálásra belsőégésű motorok összeszerelésénél

Machine Learning Applications for Fault Detection in Internal Combustion Engine Assembly

Szűcs Balázs

Széchenyi István Egyetem, Győr, Magyarország
szucsbalazs90@gmail.com

Összefoglalás — A gépi tanulás manapság egyre nagyobb teret hódít a hétköznapi életben. Az egészségügyi alkalmazásokon át, gépi látásban, ajánló rendszerekben, különböző virtuális asszisztensekben, beszédfelismerő, beszédszintetizáló, fordító alkalmazásokban, valamint a leghétköznapibb dolgokban is találkozhatunk gépi tanulás és mesterséges intelligencia algoritmusokkal. A mindennapi alkalmazásokon túl egyre nagyobb hangsúly kerül az ipari felhasználásra. A kutatás-fejlesztésben, az anyagtudományban, robotikában, illetve az Ipar 4.0 terjedésének és a digitalizációnak köszönhetően a gyártásban is egyre több gépi tanulásra épülő rendszer kerül bevezetésre. A különböző érzékelők, mérőátalakítók és mérőberendezések által előállított nagy adathalmaz kiváló kiindulási alapot biztosít a berendezések és termékjellemzők vizsgálatára, nem ismert összefüggések feltárására. A tanulmányban példákon keresztül bemutatásra kerülnek a mesterséges intelligencia algoritmusok lehetséges felhasználási módjai gyártásban, valamint egy hibadetektálási eljárás, melynek következtében a motorszerelde termelékenysége növelhető, illetve a kritikus szerelési hibák elkerülhetőek. A belsőégésű motorok főtengelyeinek átfogatási nyomaték méréseit mesterséges intelligencia algoritmussal elemezve nem ismert összefüggések kerültek feltárásra, melyek alapján az összeszerelési folyamat korai szakaszában detektálható az idegen anyag jelenléte a csapágyakon és a főtengelycsapokon. A hiba korai felismerésével elkerülhető válik az utómunkázandó motorok felesleges továbbépítése.

Kulcsszavak: gépi tanulás, mesterséges intelligencia, neurális hálózatok, belsőégésű motor, hibadetektálás, digitalizáció, Big Data, ipar 4.0, gyártás, gyártásoptimalizálás, minőségbiztosítás

Abstract — Nowadays the machine learning algorithms gains more and more ground in everyday life. Through healthcare applications, machine vision, recommendation systems, various virtual assistants, speech recognition systems, speech synthesizers, translator applications and even in the most common things we can encounter machine learning and artificial intelligence algorithms. In addition to everyday usage, an increasing emphasis is placed on industrial use. In the field of research and development, materials science, robotics and thanks to the spread of Industry 4.0 and digitalization, more and more machine learning based systems are being introduced in production. The large

dataset produced by various sensors, measuring transducers and measuring devices provides an excellent starting point for examining equipment and product features, and to find any unknown connections between features. This paper gives examples of possible ways of using machine learning algorithms in manufacturing, as well as an error detection method, which can lead to increased production rate and avoid critical assembly errors. Measurements of the crankshafts turning torque of internal combustion engines and inspection them by machine learning algorithm revealed unknown correlations that allow the foreign material to be detected on the bearings and the crankshafts at an early stage of the assembly process, thus unnecessary further work can be avoided.

Keywords: machine learning, artificial intelligence, neural networks, internal combustion engine, fault detection, digitalization, Big Data, Industry 4.0, manufacturing, manufacturing optimization, quality assurance

1 BEVEZETÉS

Napjainkban egyre nagyobb hangsúly kerül a különböző gépi tanulási algoritmusokra és számítási intelligencia megoldásokra épülő eszközökre és szolgáltatásokra. Az internet széleskörű elterjedésének, a közösségi platformok népszerűségének és az egyéb online szolgáltatásoknak hála rengeteg adat áll a rendelkezésünkre, melyek a mesterséges intelligencia algoritmusok fejlesztéséhez és tanításához elengedhetetlen. Ennek következtében elsősorban az online világban, valamint a szórakoztató és fogyasztói elektronikai eszközökben terjednek ezek a megoldások a leggyorsabban. Az Ipar 4.0 terjedésének köszönhetően azonban egyre nagyobb lehetőségek tárulnak fel a mesterséges intelligencia (MI) ipari felhasználására is. A terepi szinttől a vállalatirányítási rendszerekig egyre több és több eszköz kapcsolódik vállalati hálózatokhoz. Az IoT (*Internet of Things, a dolgok internete*) eszközök, különböző mérőberendezések és szenzorok a nap huszonnégy órájában generálják az adatokat, és az így létrejövő adathalmaz, egy úgy nevezett *Data Lake* kiváló alapot kínál az MI módszerekkel való kísérletezésre.

A belső égésű motorok és különböző alkotó elemeinek gyártása, valamint összeszerelése során rengeteg mérési adat keletkezik. A mérési eredmények kiértékelése és

ellenőrzése elengedhetetlen a szigorú gyártási és minőségi előírások betartásához. Az autógyártásban a sorozatgyártás természete miatt, valamint a törvényi előírásoknak köszönhetően több évre visszamenőleg, hatalmas mennyiségben állnak rendelkezésre a különféle mérési és egyéb adatok, feljegyzések.

A munkámban a gyártási folyamat során keletkezett nagy adathalmazra épülő, az iparban lehetségesen használható módszerek felkutatására és gyakorlati alkalmazására összpontosítottam, különös tekintettel a főtengelyek átforgatási nyomatékok és a csapágykarral járó motorhibák közötti lehetséges összefüggések feltárására. A cikkben tehát az adott kérdéskörben alkalmazott módszerek, illetve eredmények kerülnek bemutatásra.

2 ANYAG ÉS MÓDSZER

Ebben a fejezetben a felhasznált adatok, az adatok előfeldolgozása, a fejlesztői környezet és az alkalmazott gépi tanulási módszerek kerülnek bemutatásra, mint a **K-átlag**, **főkomponens analízis (PCA)**, **lineáris diszkriminancia-analízis (LDA)**, **DBSCAN** és az **LSTM neurális hálózatok**. A k-átlag, a PCA, az LDA és a DBSCAN a felügyelet nélküli gépi tanulási módszerek közé tartozik, melyeket elsősorban adatbányászatra használják, az LSTM pedig használható felügyelet és felügyelet nélküli tanulásra is.

A felsorolt módszerek, eredményeik és egymásra épülésük a 3. fejezetben kerülnek bemutatásra.

2.1 A fejlesztői környezet

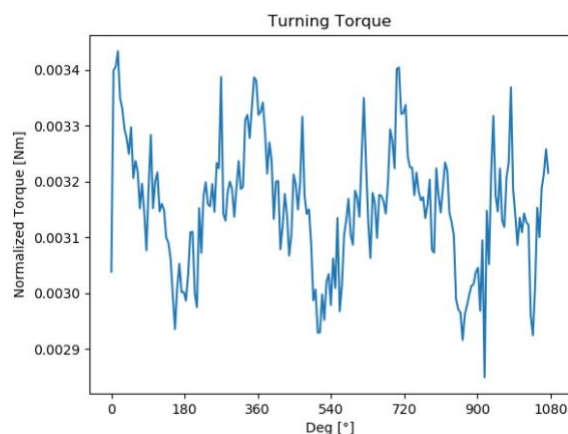
Az adatok előfeldolgozására és az algoritmusok implementálására egyaránt a gépi tanulásban egyre nagyobb népszerűséget örvendő programnyelvet, a Python 3.6.6-os verzióját használtam. Az adatok formázására, előfeldolgozására és megjelenítésére a numpy [1], a pandas [2] és a matplotlib.pyplot [3] modulokat, illetve a gépi tanulásához és a klaszterezéshez a scikit-learn 0.20.0 [4] és a TensorFlow 1.12 [5] csomagokat hívtam segítségül.

2.2 A felhasznált adatok

Munkám során egy motorszerelde terepi méréseit és egyéb reklamációs, valamint minőségbiztosítási feljegyzéseit elemeztem és értékeltem ki a gépi tanulási módszerek segítségével.

A gyártási folyamat egy szakaszában a motorblokkba beépítésre kerülnek a főtengelycsapágyak, a főtengely, a hatókarok, hajtókarcsapágyak, dugattyúk és egyéb alkatrészek, tömitések. A szerelési folyamat végeztével egy automata mérőberendezés száz százalékban megméri az elkészült motorokban a főtengelyek átforgatási nyomatékát. A mérés során a berendezés rákapcsolódik a főtengelyre és egy villamos motor segítségével megforgatja azt. Forgatás közben a villamos motor áramfelvételéből kiszámítja az egyes szöghelyzetekhez tartozó nyomatékértékeket. A mérés közben három teljes körbefordulás során, 5,4 fokként, összesen kétszáz ponton kerülnek az adatok tárolásra. A mérőberendezés csak azt vizsgálja, hogy a nyomaték a mérés során egy minimum-maximum toleranciaablakon belül maradjon, a görbe lefutásából, jellegéből nem von le következtetést. A mérés során a szöghelyzet-nyomaték párokon túl tárolásra kerül a motor egyedi azonosítószáma, a mérés időpontja, a mérőprogram száma, a mérésre vonatkozó diagnosztikai adatok és a mérés kimenetelének eredménye.

A kutatáshoz a 2016-os és, egy adott motortípushoz tartozó mérési eredményeit használtam fel. Az előfeldolgozás után összesen **141 351 mérés** került kiértékelésre. Egy tipikus átforgatási nyomaték mérés eredménye az 1. ábrán látható.



1. ábra: Tipikus átforgatási nyomaték görbe

2.3 Az adatok előfeldolgozása

Az adatok előfeldolgozása során szükség volt az ismétlődő adatok, a hiányos és hibás bejegyzések eltávolítására, valamint a táblázatok megfelelő formára hozására [6]. Az adatok vizsgálata során csak a „jó” eredményű mérésekkel folytattam a munkát, tehát a mérések azon halmazát vizsgáltam, mely tartalmazza ellenőrzésen potenciálisan túljutó hibás motorokat.

Az előfeldolgozás során az adatok normalizálására [7][8] is kerültek. A 200 dimenziós bemeneti vektorokat egyenként egységnyi normájúvá skáláztam a scikit-learn és az L2-norma segítségével.

2.4 K-átlag

Az adatokban fellelhető mintázatok, csoportok felderítésére a K-átlag algoritmust egy továbbfejlesztett változatát, a k-átlag++ algoritmust használtam [9]. A k-átlag úgy osztályozza a mintákat, hogy azokat C különálló csoportba sorolja úgy, hogy a klaszteren belüli távolságok négyzetösszegét, az úgynevezett tehetetlenséget minimalizálja:

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (1)$$

Minden klasztert az átlagos μ_j jellemezi, melyet centroidnak hívnak. A tehetetlenség, melyet szokás inerciának is nevezni, egy nem normalizált mérték: csak azt tudjuk, hogy az alacsonyabb értékek jobbák és a nulla az optimális. Ezt kihasználva határoztam meg az adatokban fellelhető és elkülöníthető csoportok ideális számát. Iteratív módon növeltem a feltételezett csoportközéppontok számát, majd minden egyes lépésben az algoritmus kiszámolta a teljes mintára vonatkozó inerciát. Addig növeltem a csoportok számát, míg az inercia nem teljesítette azt a kritériumot, hogy egy újabb csoport hozzáadása esetén számottevően már nem csökken az inercia értéke.

Fontos megjegyezni, hogy nagyon magas dimenziójú terekben az euklideszi távolságok inkább felfűjődnek (ez az úgynevezett „dimenziós átok”). Egy dimenziócsökkentő algoritmus, például a PCA futtatása a k-átlag algoritmus

előtt enyhítheti ezt a problémát, valamint felgyorsíthatja az algoritmus futását.

2.5 PCA

A gépi tanulásban és adatbányászatban a dimenziócsökkentés és információ kinyerés egyik lehetséges módja a PCA (Principal Component Analysis).

A PCA vagy főkomponens analízis [10] egy többváltozós statisztikai eljárás. A PCA lényege, hogy egy nagy adathalmazt, melynek változói kölcsönös kapcsolatban állnak egymással, dimenzióit lecsökkente, miközben az adathalmaz varianciáját a lehető legjobban megtartja. A dimenziócsökkentést úgy hajtja végre, hogy egy ortogonális transzformáció segítségével az adathalmaz lehetségesen korreláltható változóit lineárisan korrelálatlan változók értékészletévé alakítja át. Ezeket a komponenseket főkomponenseknek nevezünk. A főkomponensek száma kisebb vagy egyenlő az eredeti változók számával.

Az adatok gyorsabb feldolgozásához és egyszerű ábrázolásához a PCA-t alkalmaztam, mint dimenziócsökkentő eljárás. A 200 dimenziós vektorokból 2 dimenziós vektorokat készítettem úgy, hogy a két legnagyobb varianciájú főkomponenst tartottam meg. Ezeket az adatpárokat aztán egy pontdiagramon ábrázoltam, illetve segítségükkel meghatároztam a klaszterek ideális számát.

A pontdiagramon egyes egyedek jellegzetes eltérést mutattak az átlagos mintákhoz képest. Ezek a pontok a csoportjaik középpontjától távol, szórta helyezkednek el.

2.6 Lineáris diszkriminancia-analízis

A PCA egyik alternatívája, a vele közeli rokonságban álló lineáris diszkriminancia-analízis (LDA) [11]. Az LDA a független változók olyan lineáris kombinációját képes megtalálni, amely a függő változó alapján kialakított csoportokat a lehető legjobban megkülönbözteti.

Az LDA a diszkriminálásnak köszönhetően a térben jobban elkülöníti az egyes pontokat, ezért az adatok ábrázolásánál az alkalmazása célszerűbb lehet, mint a PCA használata. A módszer segítségével pontosabban sikerült feltárni az egyes csoportokban rejlő szélsőséges egyedeket, melyek a későbbi vizsgálatok alapját képezik.

Az LDA első lépése a diszkrimináló függvény(ek) kiszámítása. A függvények száma úgy számítható ki, hogy a függő változó lehetséges kimeneteleinek száma -1 és a független változók számának a minimumát vesszük. A diszkrimináló függvény általános:

$$D_j = d_0 + d_{1j}x_1 + d_{2j}x_2 + \dots + D_{kj}x_k \quad (2)$$

ahol j az adott függvény sorszáma, az x_i -k a mért független változók, d_0 konstans, a d_{ij} a x_i mért változók j -edik diszkrimináló függvényhez tartozó együtthatója. A függvény akkor optimális, ha a függő változó által meghatározott csoportok közötti külső négyzetösszeg és a csoportokon belüli négyzetösszeg hányadosa maximális. Az algoritmus hátránya, hogy szükséges megadni, hány csoportba kívánjuk az adataikat sorolni, ezért alkalmazása előtt szükséges felderíteni a mintában rejlő csoportokat, például a k -átlag és a könyök módszer segítségével.

Az LDA algoritmus a PCA-val együtt a scikit-learn csomagban került megvalósításra.

2.7 Szélsőséges egyedek vizsgálata

Az adatokban feltárt szélsőségek elemzése során arra lettem figyelmes, hogy a nyomatókgörbék lefutása bizonyos esetekben jellegzetes képet mutat.

Mivel nagymennyiségű adattal dolgoztam a szélsőséges egyedek manuális szelektálása és vizsgálata rendkívül időigényes feladat lett volna, ezért szükségem volt egy megoldásra, amely segítségével automatikusan kiválogathattam a feltűnő egyedeket.

Az adatok gépi felcímkézésére a DBSCAN algoritmust választottam, a hibás egyedek detektálására pedig az LSTM neurális hálózatokat.

2.8 DBSCAN

A k -átlag, a PCA és az LDA algoritmusok eredményeinek a további vizsgálatához a gépi tanulásban az egyik legelterjedtebb klaszterező algoritmust, a DBSCAN-t használtam fel.

A DBSCAN, azaz a Density Based Spatial Clustering of Applications with Noise [12] egy sűrűség alapú klaszterezési eljárás. A térbeli pontthalmazt a pontok egymáshoz viszonyított elhelyezkedésük (térbeli sűrűségük) alapján csoportosítja. A térben sűrűn elhelyezkedő, egymáshoz közeli pontokat egy klaszterbe sorolja, míg a térben ritkábban, a sokaságtól távol elhelyezkedő pontokat zajként osztályozza az algoritmus. Az eljárás egyik előnye, hogy nem szükséges megadni a keresett csoportok számát, csupán azt az epszilon távolság határértéket, mely az egyes csoportokon belül a pontok egymáshoz viszonyított maximális távolságát adja meg. Az epszilon megfelelő megválasztásával automatikusan zajként osztályozhatóak az anomáliát tartalmazó egyedek, melyek az osztályuk középpontjától távol, ritkábban helyezkednek el.

A k -átlag és az LDA algoritmusok által feltárt csoportokat egyenként vizsgáltam a DBSCAN segítségével. Az eljárás a scikit-learn csomagban implementálásra került, működésének részletes leírása Martin et. al [12] munkájában olvasható.

2.9 Long Short-Term Memory (LSTM) hálózat

A munkám során vizsgált adatoknak kevesebb, mint 2%-a tartalmaz jellegzetes anomáliát, ezért klasszikus neurális háló alapú osztályozáshoz nem áll rendelkezésre elegendő „rossz” példa. A hálózat, ha az összes rossz egyedet jónak osztályozná, akkor is 98%-os pontosságot érne el.

A dimenziócsökkentő és klaszterező eljárásokkal feltárt anomáliákat egy többretegű LSTM alapú idősor előrejelzéssel detektáltam [13]. A hálózatot csak a jó mintákkal tanítottam, így ha a bemenetere egy jó mintához tartozó idősort kapcsolunk, azt az algoritmus relatíve kis hibával, időben eltolva rekonstruálja azt kimenetén. A rossz mintákhoz tartozó idősorok esetén a rekonstrukciós hiba nagy lesz, így az eredeti és a rekonstruált minta közötti hibát kiszámítva a rossz egyedek detektálhatóak.

Az LSTM [14] vagy Long Short-Term Memory hálózat a rekurrens neurális hálózatok (RNN) egy típusa. Az LSTM hálózatok idősorok osztályozására, előrejelzésére és autoenkodernek használják.

Az LSTM elemi egysége valójában önmaga is egy többretegű neurális hálózat, egy továbbfejlesztett RNN. A módosítások elsődleges célja, hogy orvosolja az RNN-ek esetében fennálló, úgynevezett „eltűnő”, illetve „felrobbanó gradiens” problémát [15], mely a hálózatok

rekurrens jellegéből adódik. Az RNN hálózatok tanításakor a visszaterjesztett hiba a belső visszacsatolások miatt exponenciálisan csökkenhet, illetve nőhet, ezért a hálózatok tanítása rendkívül időigényes lehet, bizonyos esetekben pedig nem is konvergál. Az RNN esetében szintén probléma, hogy a hosszabb távú függőségeket a hálózat nem tudja kezelni.

A probléma megoldására az LSTM egységekben bevezetésre került egy memória egység, mely szekvenciális tanuláskor a múltbéli események eltárolása miatt képes a hosszú távú függőségek megtanulására, illetve segítségével kezelhetővé vált a gradienssel kapcsolatos említett tanítási problémák is. A memóriaegység a bemeneti kapuja segítségével képes eldönteni, hogy a visszacsatolt információt hasznos-e és megtartsa, vagy törölje azt.

Az LSTM hálózatoknak az 1997-es bemutatása óta több továbbfejlesztett változata készült, munkám során viszont a TensorFlow-ban implementált eredeti hálózatot használtam. A hálózat működésének részletes leírása Sepp Hochreiter és Jürgen Schmidhuber munkájában [14] olvasható.

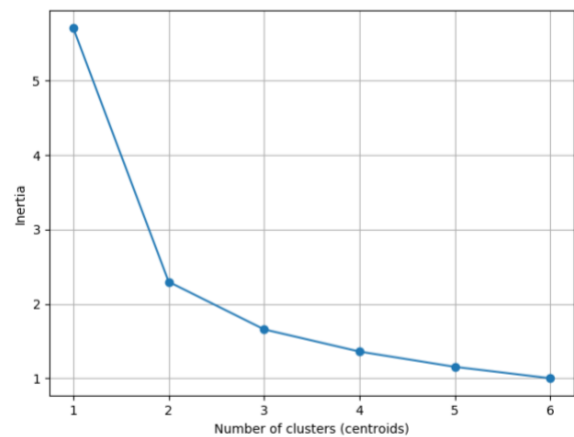
3 EREDMÉNYEK BEMUTATÁSA ÉS ÉRTÉKELÉSE

A harmadik fejezetben az általam felhasznált módszerek eredményei és értékelésük lépésről lépésre kerülnek bemutatásra. A nyers adatok csoportosításától, az anomáliák felfedezésén át jutunk el a hibás mérések detektálásáig. Az adatok tisztítására részletesen nem térek ki, az alkalmazott metodika a hivatkozott irodalomban [6] olvasható.

3.1 Klaszterek számának meghatározása a PCA és a k-átlag algoritmus segítségével

A klaszterek számának meghatározása a 2.4 fejezetben bemutatott k-átlag algoritmussal és az úgynevezett könyök módszerrel történt. Az iteratív eljárás során az algoritmus az minden iterációs lépésben növelte a centroidok számát és a (1) kifejezést minimalizálta. A kifejezés értékének alakulása a klaszterszám függvényében 2. ábrán, értékei pedig az 1. táblázatban láthatóak.

Az 2. ábráról és az 1. táblázatból is leolvasható, hogy a harmadik centroid hozzáadása után a görbe meredeksége csökkenő tendenciát mutat. Ugyan az inercia egy mértékegység nélküli metrika, de az elmondható róla, hogy a nullához közeli értékek jobbakké. Célunk tehát minél jobban közelíteni a nulla értéket. Fontos viszont megjegyezni, hogy nem minden esetben célravezető a nulla körüli tehetetlenség elérése. Ha a minta nagy szórással rendelkezik (mint esetünkben), további centroidok hozzáadásával az inercia csökkenthető, ám a valóságban az történik, hogy minden egyes kilógó, szélsőséges egyedhez egy saját csoportot illesztünk.



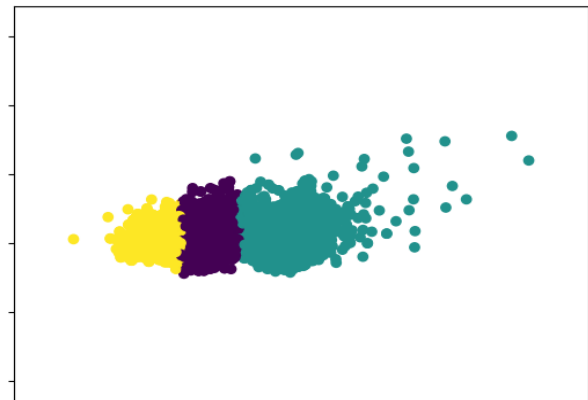
2. ábra: Az inercia a centroidok függvényében

1. táblázat: a centroid-inercia értékpárok

Centroidok száma	1	2	3	4	5	6
Inercia	5.7099	2.2981	1.6593	1.3609	1.1554	1.0023

Gyakorlati alkalmazásokban tehát célszerű azt a klaszterszámot választani, mely felett az inercia csökkenésének mértéke már erőteljes lassulást mutat.

Az algoritmus gyorsabb futása és az adatok könnyebb ábrázolása érdekében a 2.5 fejezetben bemutatott PCA algoritmust használtam. A klaszterezés eredménye a 3. ábrán látható.

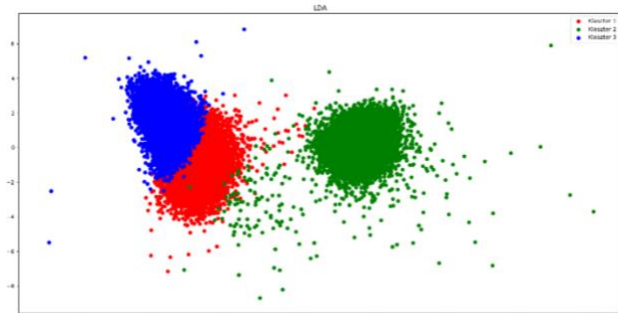


3. ábra: a PCA dimenziócsökkentés és a k-átlag klaszterezés eredménye

Az ábrázolás során a kétdimenziósra csökkentett minta értékpárjait jelenítettem meg. Az x tengelyen az első főkomponens, az y tengelyen pedig a második főkomponens került ábrázolásra, a koordináták dimenzió nélküliek. A 3. ábrából már sejthető, hogy az egyes csoportok szórása elég nagy is lehet, ezért a csoportok radikálisabb szétválasztásához az LDA eljárást alkalmaztam.

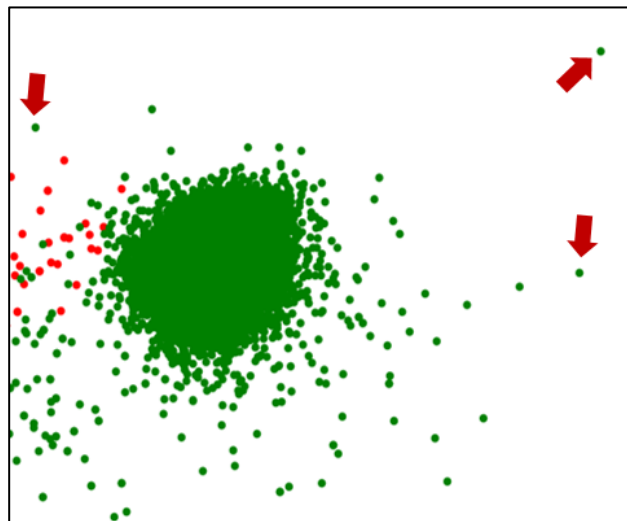
3.2 Klaszterek további vizsgálata LDA segítségével

Az LDA algoritmus egyik bemeneti paramétere a diszkriminálendő csoportok száma. Az előző fejezetben bemutatott alapján ezt a paramétert a mintában jelenlévő klaszterek száma alapján háromnak választottam. A lineáris diszkriminancia-analízis eredménye a 4. ábrán látható.



4. ábra: az LDA eredménye

Az LDA eredményének ábrázolása után az egyes csoportok szórása még szembetűnőbb. A csoportok vizsgálata során megállapítottam, hogy az adott motortípuson belüli három különálló csoport az azonos, de különböző beszállítók által gyártott, úgynevezett helyettesítő alkatrészeknek köszönhető. Ezen alkatrészek paraméterei minimálisan eltérőek lehetnek. A továbbiakban az a csoportjaik középpontjától távolabb elhelyezkedő egyedek (5. ábra) vizsgálatával folytattam a munkát.



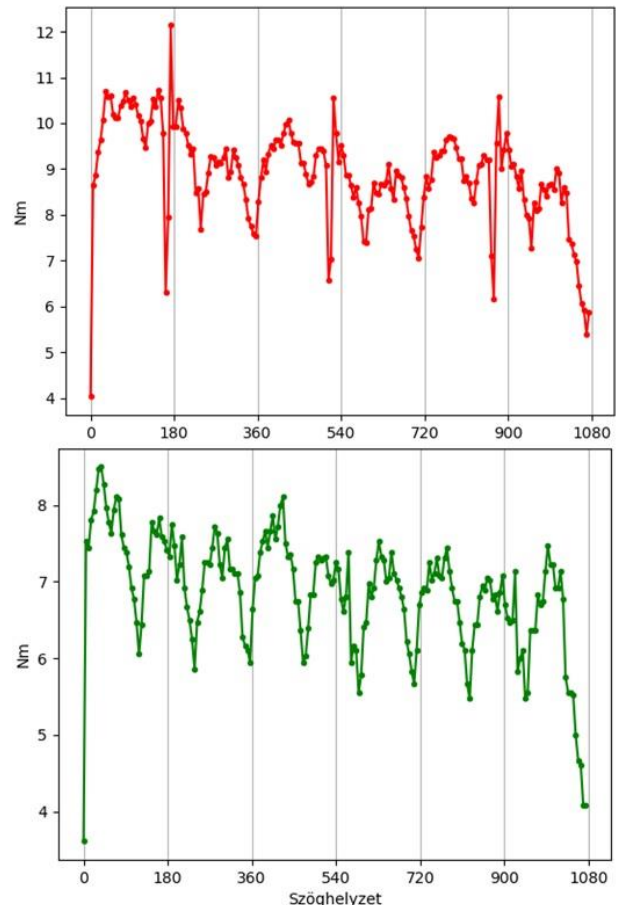
5. ábra: szélsőséges egyedek a mintában (piros nyíljal jelölve a példák)

3.3 A szélsőséges egyedek vizsgálata

A szélsőséges egyedek vizsgálata során ábrázoltam az eredeti, dimenziócsökkentés előtti mérési értékeket, majd ezeket összevettem a csoportok közepéből származó, átlagos egyedekkel.

Az ábrázolás során feltűnt, hogy míg az átlagos mérések nyomatókgörbéjének lefutása a kezdeti megindítási nyomatóktól eltérően egyenes, periodikusan csökkenő jellegű, addig a szélsőséges egyedekhez tartozó mérések esetén az adott példány átlagos mérési értékeihez képest

extrém nagy, impulzusszerű ugrásokat, kilengéseket tartalmaz a mérési görbe. Az átlagos és a feltűnő egyedek méréseinek összehasonlítása a 6. ábrán látható.



6. ábra: Egy feltűnő (piros) és egy átlagos (zöld) nyomatókgörbe összehasonlítása

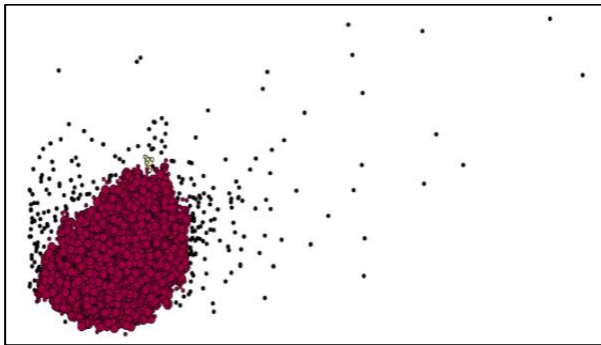
A munkát a szélsőséges egyedek beazonosításával és további vizsgálatával folytattam.

3.4 Feltűnő egyedek vizsgálata

A beazonosítás után megvizsgáltam a feltűnő példányok egyéb méréseit és amennyiben volt, utómunka jegyzőkönyveit. A mérési jegyzőkönyvek vizsgálata során kiderült, hogy a következő munkalépésben végrehajtott mérés, a főtengely axiális irányú játéknak mérése az esetek nagy részében nem megfelelő, túrésen kívüli eredménnyel zárult. A rossz főtengely axiális irányú játékkal rendelkező motorokat minden esetben javításra kerülnek, mely során az esetleges feltűnőségeket, észlelt hibákat jegyzőkönyvbe veszik. A jegyzőkönyvek elemzése során kiderült, hogy a jellegzetes nyomatókgörbével és nem megfelelő főtengely axiális irányú játékkal rendelkező motorok egy részében idegen anyagot, forgácsot vagy egyéb szennyeződést találtak a főtengely, illetve a hajtókarcsapágyakban vagy csapágyhelyeken.

3.5 A feltűnő egyedek szelektálása DBSCAN eljárással

A feltűnő egyedek manuális szelektálása rendkívül időigényes és aprólékos feladat lenne, ezért szükséges a folyamat automatizálása, melyhez a DBSCAN algoritmust használtam.



7. ábra: a DBSCAN eredménye, az átlagos mérések pirossal, a rossz mérések feketével ábrázolva

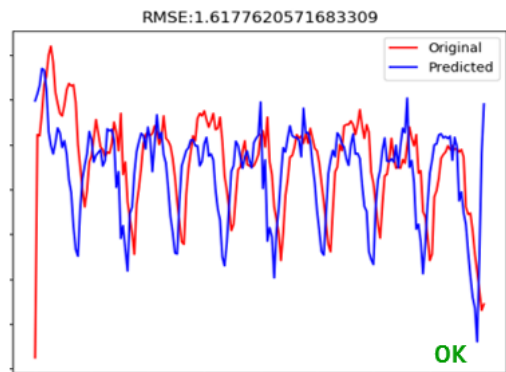
A paraméterek megválasztása tapasztalati úton történt, majd az egyes csoportokat külön-külön elemezve szétválasztottam az átlagos és a jó méréseket. Az eljárás során kihasználtam a DBSCAN azt a tulajdonságát, hogy a ritkábban elhelyezkedő egyedeket egységesen zajként osztályozza, így tehát két csoportra tudtam bontani a mérési eredményeket. Az algoritmus szemléltetése a 7. ábrán látható.

A minták osztályozása után már lehetővé vált a hibadetektáló neurális hálózat tanítása.

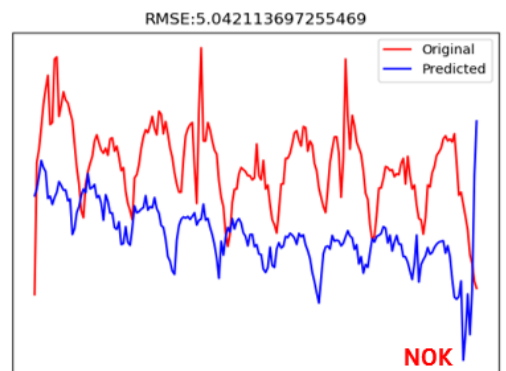
3.6 Hibadetektálás LSTM neurális hálózattal

A neurális hálózatok felügyelt tanításához szükséges a tanító adatok felcímkézése. Az adatok osztályozása is a felügyelt tanítás egy ágazata, azonban esetünkben a megkülönböztetni kívánt osztályok elemszáma nagymértékben eltér. A teljes minta csupán 2%-át teszik ki a DBSCAN algoritmus által rossznak osztályozott mérések, ezért a 2.9 fejezetben bemutatott LSTM idősor előrejelzés segítségével azonosítottam a hibás egyedeket [13].

A többrétegű LSTM neurális hálózat tanításakor a csak a jó mérési eredményeket használtam fel. Az eredeti mintákból öt időlépéses eltolással új tanítómintákat hoztam létre. Az így kapott $5 * 200$ dimenziós egyedekkel feltanított hálózat kellően nagy pontossággal tudja rekonstruálni a bemenetére érkező jel időbeli eltolját a kimenetén. Mivel a hálózat a hibás egyedekkel nem találkozott a tanítás során, ezért a bemenetére érkező hibás méréseket csak nagy hibával tudja rekonstruálni a kimenetén. A be- és kimeneti jeleket egymáshoz igazítva szembevetendő az eltérés. A potenciálisan idegen anyagot tartalmazó motorok azonosításához így elegendő kiszámolni a rekonstrukciós hibát, például az átlagos négyzetes hibát. A LSTM rekonstrukció eredménye, illetve a számított átlagos négyzetes hiba (RMSE) a 8., illetve a 9. ábrán látható.



8. ábra: jó mérési (piros) görbe és rekonstruáltja (kék), RMSE: 1.6178



9. ábra: rossz mérési (piros) görbe és rekonstruáltja (kék), RMSE: 5.0421

4 KÖVETKEZTETÉSEK

A fent bemutatott módszerek és eredményeik alapján kijelenthető, hogy már a legegyszerűbb gépi tanulási módszerek is sikerrel alkalmazhatóak az iparban. A mérési eredményekben rejlő mintázatok és nem ismert összefüggéseik feltárására segítségével a gyártási folyamat optimalizálható, a költségek csökkenthetőek.

A klaszterező eljárások segítségével feltárt csoportok rávilágítottak az elméletben azonos, ám a gyakorlatban különböző paraméterekkel rendelkező, helyettesítő alkatrészek közötti különbségekre. Ezek az eredmények remek kiindulási lehetőséget jelentek a későbbi termék- és gyártásoptimalizáláshoz.

Az adatok csoportosítása és ábrázolása felismerhetővé tette, hogy ugyan az egyes mérések értékei a toleranciasávon belül mozognak, ám bizonyos egyedeknél a nyomatók görbe lefutásának jellege jelentősen eltérhet az átlagos mérési eredményekétől. Ezek a szélsőséges egyedek, illetve mérési eredményeik indikátorai lehetnek egy későbbi hibás mérésnek vagy akár a motor meghibásodásának. Az eredmények kiértékelésének optimalizálásával csökkenthető a selejtarány, az utómunka okozta idővesztés és az egyéb költségek.

A szélsőséges egyedek automatikus szelektálására kiváló lehetőséget nyújt a DBSCAN algoritmus. Használata egyszerű, nem igényel előzetes ismeretet az adathalmazról. Alkalmazása segítségével az adatok könnyen felcímkézhetőek, ezáltal a felügyelt tanulás első lépéseként kiválóan alkalmazható szabálytalan, többdimenziós adatok esetén.

A feltárt és szelektált szélsőséges egyedekkel kapcsolatos adatok feldolgozásának egyik lehetséges buktatója az emberi tényező. A jegyzőkönyvek és

hibaleírások hanyag, illetve hiányos kitöltése rendkívüli módon megnehezítheti az adatok kiértékelését és az összefüggések feltárását. Az emberi adatgyűjtés tehát szabályozott kereteket, illetve rendkívüli fegyelmet és odafigyelést igényel. A standardok kialakítása és betartása fontos az adatrögzítésnél is.

Az LSTM hálózatok egy lehetséges alkalmazási lehetősége az idősorok előrejelzésére és osztályozására. Ha viszonylag kevés minta áll rendelkezésre egy adott csoportból, a fent bemutatott módszer segítségével könnyen osztályozhatjuk az egyes méréseket. A potenciálisan gyanús mérések korai észlelése segít az idegen anyag okozta csapágykáros motor kiesések elkerülésében, így minőségi javulás és költségcsökkenés érhető el.

A gépi tanulási módszerek tehát hatalmas potenciállal rendelkeznek az autóipar és az egyéb ipari szektorok területén is. A fent bemutatott és további számítási intelligencia módszerek alkalmazási területei és lehetőségei további kutatásokat igényelnek a témában.

5 ÖSSZEGZÉS

Munkám során egy belső égésű motorokat gyártó szereldei gyártósor mérései eredményeit elemeztem gépi tanulási módszerekkel. A kutatás célja a főtengely átforgatási nyomatékának mérése és a csapágykarral járó motorhibák közötti összefüggések feltárása volt. A kutatásban csak a jónak értékelt mérési eredményeket használtam fel.

Az adatok mesterséges intelligenciával történő elemzéséhez az elsődleges feladat az adatok szelektálása és rendezése volt. Ez a gépi tanulási módszerek egy sarkalatos pontja. Az előfeldolgozás további lépéseiben szükséges volt az adatok normalizálása és a dimenziócsökkentés is, melyek segítségével a mesterséges intelligencia algoritmusok teljesítménye és pontossága javítható, tanítási sebessége gyorsítható.

Az adatok szűrése és előfeldolgozása után a mintát különböző klaszterező eljárásokkal vizsgálva meghatároztam a mintában fellelhető, az adott motortípuson belüli különböző csoportok számát. Az egy típuson belüli alcsoportok, melyek a különböző helyettesítő alkatrészekkel szerelt motorokat reprezentálják, megállapítottam, hogy a csoportjaiktól távol eső, szélsőséges egyedek mérési görbéi eltérést mutatnak az átlagostól.

Az egyes egyedeket azonosítottam és a nyomatékgörbéket a reklamációs és utómunka jegyzőkönyvekkel, valamint egyéb mérésekkel összehasonlítva megállapítottam, hogy az esetek nagy részében a főtengely axiális irányú játéka mérés nem megfelelő, illetve a motorok utómunkája során a főtengely, vagy a hajtókarcsapágyakban idegen anyagot találtak. A feltárt jelenség ismeretében megtanítottam egy LSTM neurális hálózatot a jó mérési görbék pontos rekonstrukciójára. A hálózat a rossz mérési görbéket viszont nagy hibával rekonstruálja, így a rekonstrukciós hiba kiszámításának a segítségével a hibás mérések valós időben, automatizált módon detektálhatóvá váltak.

A feltárt összefüggések és módszerek segítségével tehát az idegen anyag jelenléte a belső égésű motorok főtengely- és hajtókarcsapágyaiban előre jelezhető, így jelentős minőségjavulás és költségcsökkentés érhető el, épp ezért a

mesterséges intelligencia további ipari alkalmazásai és a módszerek tökéletesítése további kutatásokat igényel.

IRODALOMJEGYZÉK

- [1] Travis E, Oliphant. (2006), A guide to NumPy, *USA: Trelgol Publishing*, <http://web.mit.edu/dvp/Public/numpybook.pdf>
- [2] Wes Mckinney (2011), pandas: a Foundational Python Library for Data Analysis and Statistics, <https://www.scribd.com/doc/71048089/pandas-a-Foundational-Python-Library-for-Data-Analysis-and-Statistics>
- [3] Hunter et al (2007), Matplotlib: A 2D graphics environment, *Computing in Science & Engineering, Volume 9, Number 3*, 90-95, <https://ieeexplore.ieee.org/document/4160265>
- [4] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, *JMLR* 12, pp. 2825-2830, <https://dl.acm.org/citation.cfm?id=1953048.2078195>
- [5] Martín et al. (2015), TensorFlow: Large-scale machine learning on heterogeneous systems, <https://arxiv.org/abs/1603.04467>
- [6] Hadley Wickham (2014). Tidy Data, *Journal of Statistical Software*, Volume 59, Issue 10., <https://www.jstatsoft.org/article/view/v59i10/v59i10.pdf>
- [7] Vafaei, Nazanin & Ribeiro, Rita & Camarinha-Matos, Luis. (2015). Importance of Data Normalization in Decision Making: case study with TOPSIS method., https://www.researchgate.net/publication/304539326_Importance_of_Data_Normalization_in_Decision_Making_case_study_with_TOPSIS_method
- [8] S. B. Kotsiantis, D. Kanellopoulos and P. E. Pintelas (2006), Data Preprocessing for Supervised Learning, *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE VOLUME 1 NUMBER 1 2006* ISSN 1306-4428, https://www.researchgate.net/publication/228084519_Data_Preprocessing_for_Supervised_Learning
- [9] David Arthur, Sergei Vassilvitskii (2007), k-means++: The Advantages of Careful Seeding, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, <https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf>
- [10] Pearson, K. (1901). „On Lines and Planes of Closest Fit to Systems of Points in Space” . *Philosophical Magazine* 2 (11), 559–572. , <http://pca.narod.ru/pearson1901.pdf>
- [11] McLachlan, G. J. (2004). Discriminant Analysis and Statistical Pattern Recognition. *Wiley Interscience*. ISBN 0-471-69115-1. MR 1190469., <https://pdfs.semanticscholar.org/20ce/95262aa2781c2c3127ca77f18afce3c8f69.pdf>
- [12] Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M., eds. *A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press. pp. 226–231. <https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf>
- [13] P Malhotra, L Vig, G Shroff, P Agarwal (2015), Long Short Term Memory Networks for Anomaly Detection in Time Series, *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8*. Available from <http://www.i6doc.com/en/>
- [14] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". *Neural Computation*. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276, <http://europepmc.org/abstract/MED/9377276>
- [15] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks*, 5(2), 157–166., <http://ai.dinfo.unifi.it/paolo/ps/tnn-94-gradient.pdf>