# Kutatásmódszertani Terv

**A Szent István Egyetem Gazdálkodás és Szervezéstudományok Doktori Iskola 2018. évi komplex vizsgára**


**Barta Gergő**

**II. éves PhD hallgató**

**Neptun kód: N787OJ**

**Témavezető: Pitlik László**


**2018. június 11.**

# Tartalomjegyzék

# 1. Bevezetés

A 2016-os évben a doktori képzés központi reformon ment keresztül, melynek egyik eredményterméke a komplex vizsga bevezetése, melyet a hallgatóknak a második év végén kell teljesíteniük, bizonyítva megfelelő felkészültségüket és szakmai ismeretüket. A komplex vizsga szerves részét képezi a Disszertációs Rész megírása mellett, a Kutatásmódszertani Terv (a tételes vizsga keretében), mely célja a Kutatásmódszertan és a Gazdaságelemzési Módszerek című tárgyak ismereteire építkezve kifejteni a kutatásban alkalmazni kívánt módszereket, valamint a primer és/vagy a szekunder adatok körét és forrásait és a mintaválasztási elképzeléseket. A dokumentum a felvázolt pontok bemutatását célozza meg, ezenkívül részletezi a kutatás teljes folyamatát, alkalmazandó modelleket és jóságmetrikákat.

A tervezett értekezés jelen címe:

**Mesterséges intelligenciák alkalmazása az informatikai rendszerek biztonsági auditjában**

A Disszertációs Rész keretében feltárt (legitim, racionális innovációt sejtető) hipotézisek az alábbiak:

**H1:** Különböző logikát/megközelítést alkalmazó mesterséges intelligencia módszertanok magasabb fokú hibridizációjával magasabb teljesítmény érhető el az incidensek/anomáliák detektálására vonatkozóan.

**H2:** A kutatásban elkészítendő egyes hibrid modellek képesek az előre definiált normális tevékenységek között meghúzódó összefüggések matematikai leképezésére, s ez által az anomáliák feltárására, kísérletekkel alátámasztott optimális parametrizálásával azok megtanulására a túlilleszkedést minimalizálva.

**H3:** A kutatásban elkészítendő hibrid modellek a többváltozós matematikai statisztikai módszerekkel szemben meghaladják az előzetesen definiált jóság metrikák aggregált szintjét.

**H4:** A kutatásban elkészítendő hibrid modellekben az input jelként felhasznált adatokat lehetséges olyan szinten transzformálni és a modellekbe lehetséges olyan védelmi mechanizmusokat beépíteni, hogy azok ellenállók legyenek egyes külső szándékos manipulációkkal szemben, melyek célja az anomália-detektáló rendszer félrevezetése, vagyis az anomáliák normális magatartásként történő feltüntetése.

A dokumentum 1. számú melléklete tartalmazza a kutatás munkamenetét bemutató folyamatábrát. A Kutatásmódszertani Terv ezen a munkameneten keresztül került kifejtésre, azaz részletezi a releváns szakirodalmak feldolgozását, az adatgyűjtést, adatbázis létrehozását, a változók kódolását, az adathalmazok felbontását, a modellek fejlesztését és optimalizálását, az eredmények összehasonlítását és a következtetések levonását a felsorolt sorrendben. A dokumentumban, tehát, annak levezetése történik meg, hogy a tervezett kutatás munkamenete racionális eredményeket képes produkálni a tervezés szintjén.

# 2. Kutatási munkaterv

## 2.1. Témához kapcsolódó szakirodalmi kutatás és elemzés

A Disszertációs Részben feltárt hipotézisek realizálhatóságát érintő kutatás célja olyan modellek megalkotása, melyek az informatikai rendszerekben felmerülő belső információbiztonsági incidensek, azaz anomáliák felderítésére irányulnak a mesterséges intelligencia alapú fogalomalkotás lehetőségeit felhasználva úgy, hogy a jelenlegi legjobb megoldások („*benchmark*” adatok) bizonyítottan meghaladásra kerüljenek. A mennyiségi meghaladáshoz a mindenkor rendelkezésre álló naplóállományokból kell a mesterséges kockázatfogalom megalkotása keretében ennek mértékét és normáját algoritmikusan levezetni. A kutatási célként feltárt hipotézisekre vonatkozó alternatív megoldások értékeléséhez nélkülözhetetlen a releváns szakirodalom tanulmányozása, mely a téma sajátossága alapján két részre bontható.

Az első rész az információbiztonságban és anomália észlelésben megjelent kutatásokra összpontosít. A témában megjelent kutatások, modellek, felhasznált adatbázisok és eredmények építőkövekként szolgálnak a saját adatbázisok létrehozásához, melyet úgy kell megtervezni, hogy elemzésük során azok alkalmasak legyenek a hipotézisek bizonyításához, a tudományos szimuláció keretében a modellalkotáshoz. A szakirodalom tanulmányozása keretet ad a szükségesnek ítélt input és output változók köréről, amelyek az elemezni kívánt adatbázist alkotják. A szimulációt vállalatirányítási rendszerekből kinyert naplóállományok alapján kívánom elvégezni, ezért szükséges, hogy bizonyítható és hitelesített módon kategorizálásra kerüljenek az adatrekordok, hogy azok önmagukban, vagy többi rekorddal összetételben, információbiztonsági incidensnek minősülnek-e vagy sem.

A második részben mesterséges intelligencia szakirodalmának áttekintése során az alkalmazni kívánt modellek és egyéb mesterséges intelligencia fogalomkörébe tartozó eszközök és algoritmusok tanulmányozása a cél, mely hozzájárul erősségüket és gyengeségüket elemezve, a későbbi modellfejlesztéshez.

## 2.2. Adatgyűjtés

A szakirodalom alapos áttanulmányozása, elemzése, kiértékelése, a szükséges változók és az egyes modellek elemzése után az adatgyűjtés a következő lépés a kutatás munkamenetében. Adatgyűjtésre vonatkozóan az alábbi kritériumokat támasztom előzetesen:

- Az adathalmazt/adathalmazokat éleskörnyezetben üzemelő vállalatirányítási rendszerekből kell beszerezni, mely garantálja, hogy az elemzés valós üzleti adatokon történik. A primer adatok forrása, tehát, Magyarországon üzemelő szervezetek, melyek belső vállalatirányítási rendszereikben eszközölt tevékenységeiket naplózzák, a naplófájlokat lementik és azokat ellenőrzik. Jelen terv a legalább 5 különböző szervezettől a szükséges adatok beszerzése.
- Az adathalmazt/adathalmazokat olyan szervezetektől szükséges beszerezni, mely akár előzetes vagy utólagos elemzést követően hitelt érdemlően képes megmondani egy adott rekordról, hogy az anomáliának számít-e vagy nem, tehát, képes a célváltozó értékét az adathalmazban hitelesen meghatározni. Ez hozzájárul ahhoz, hogy a modellfejlesztés során a normális magatartásokat és anomáliákat tartalmazó rekordok közötti összefüggések és minták kielemezhetővé váljanak úgy, hogy azok valós képet adjanak az anomália létéről. Amennyiben az előzetes kiértékelés nem lehetséges, az anomáliákat fel kell fedezni a rendszerben (pl. nem felügyelt gépi tanuló eszközökkel) és azokat a szervezettel közösen ki kell elemezni és az anomáliára utaló rekordokat meg kell erősíteni.
- Az adathalmazt/adathalmazokat olyan szervezetektől kell beszerezni, ahol a naplófájlok legalább 1 évre visszamenően elérhetőek, mely garantálja, hogy megfelelő mennyiségű adat rendelkezésre áll a modellfejlesztéshez. Ez összhangban áll Banko és Brill (2001) a Microsoft kutatóinak adatgyűjtésre vonatkozó ajánlásával.
- Az adathalmazt/adathalmazokat úgy lenne célszerű beszerezni, hogy a naplófájlok lehetőleg „csv” kiterjesztésű fájlokban, vagy „SQL” adatbázisban rendelkezésre álljanak. Amennyiben ez nem lehetséges, szükséges az állományok további konverziója, hogy azok alkalmasak legyenek adatbázisba való letárolásra.
- Az adathalmaznak/adathalmazoknak rendelkezniük kell a minden kétséget eloszlató pontos attribútum megnevezésekkel, azok leírásával és értékkészletük meghatározásával.

- Amennyiben szükséges az adathalmazokat anonimizálva kell kielemezni, hogy ne sértse a munkatársak személyes jogait.
- A beszerzett adathalmazban nem cél a mintaszerű kiválasztás, a teljes adathalmazt fel kell használni a modellalkotásban, mely a pontosabb becsléshez járul hozzá (Raschka, 2015).

Jelen terv szerint adatokat legalább 5 különböző szervezettől kívánom beszerezni a fent említett kritériumok mellett. A szervezetekkel jelenleg egyeztetés folyik az adatok átadásáról.

## 2.3. Adatbázis létrehozása

Az adatgyűjtést követően szükséges az adatok rendszerezése, strukturált formába való öntése, hogy azok alkalmasak legyenek gépi feldolgozásra. A legtöbb vállalatirányítási rendszer képes ma már strukturált fájlban naplóállományok exportjára, azonban, strukturálatlan adatok esetén szükséges annak adatbázisba való beolvasásra való felkészítése, mely manuális adatfeldolgozást igényel. Az adatbázis jelen tervek szerint Anaconda Navigator 1.7.0 fejlesztői környezetben Python 3.6 Panda keretrendszerbe kerül feltöltésre, mely lehetőséget biztosít az adatok hatékony lekérdezésére, manipulálására, logikai ellenőrzésére gépi kódolás által.

Az elkészítendő adatbázisban az input változók az egyes audit bejegyzések, melyekhez kiszolgáló információkat szolgáltatnak a felépített infrastruktúra leíró elemek és a felhasználókra vonatkozó adatok. Egy sorhoz egy célváltozó tartozik, tehát, az input változók összességének (összes rekord, összes változó) az alábbi mátrix felépítést kell követnie az adatbázisban:

$$\begin{bmatrix} {x_1}^{(1)} & \cdots & {x_j}^{(1)} \\ \vdots & \ddots & \vdots \\ {x_1}^{(i)} & \cdots & {x_j}^{(i)} \end{bmatrix}$$

ahol $x^{(i)} = \left[{x_1}^{(i)}\ {x_2}^{(i)} \ldots {x_j}^{(i)}\right]$, $i \in \mathbb{Z}$ az adathalmaz egy rekordját jelenti, és

$$x_j = \begin{bmatrix} {x_j}^{(1)} \\ {x_j}^{(2)} \\ \ldots \\ {x_j}^{(i)} \end{bmatrix}, j \in \mathbb{Z}$$

az adathalmaz egy attribútumát jelöli. A célváltozó jelölése $y, y \in \{0, 1\}$, amennyiben bináris célváltozóról beszélünk, 0 jelölheti azt a rekordot, mely normális magatartást ír le, 1, pedig azt

a rekordot, amely anomáliát tartalmaz. A cél nem kizárólag bináris célváltozó alkalmazása, hanem az egyes anomália típusok detektálása pl. illetéktelen hozzáférés, illetéktelen adatmódosítás, külső hálózatról való hozzáférés stb., így a kutatásban a célváltozó tervezetten, további értékeket fog felvenni. Az információrendszerben elkövetett incidensek, mint anomáliák detektálása érdekében az elemezni kívánt naplóállományoknak, megítélésem szerint és az információbiztonságra vonatkozó szakirodalom kutatás eredményéül, legalább a 2. mellékletben közölt attribútumokat kell tartalmazniuk, hogy az anomália ténye észlelhető és tetten érhető legyen. További kritérium, hogy a felépített adatbázisnak képesnek kell lennie idősoros adatok kezelésére.

A 2. mellékletben kidolgozott adatbázis struktúrát az alábbiakban kell értelmezni:

- Narancssárga fejléccel szerepelnek azon adattáblák (Tranzakciók, Tranzakció állapotok, Indított tranzakciók), melyek az információrendszerekben keletkezett tranzakciók leíró adatait tartalmazzák.
- A sötétzöld fejléccel megjelölt táblák (Adatbázis táblák, Tábla műveletek) a belső, üzleti felhasználású adattáblákat írják le, melyek a tranzakciókkal összeköttetésben állnak, tehát a tranzakció naplóállományainak egyértelműen hivatkozniuk kell arra a tényre, ha egy adott üzleti tranzakció adatmanipulációval (létrehozás, törlés, módosítás, hozzáférés stb.) járt az elemzett információrendszerben.
- A világoszöld tábla (Audit bejegyzések) a naplózás karakterisztikáit hivatott eltárolni, mely meghatározza egy adott adatelérés biztonsági besorolását (pl. kritikus adat, publikus adat, személyes adat stb.) illetve az audit esemény kritikusságát.
- A szürke táblák (Felhasználói információk, Felhasználó típusok) az adott tranzakció gazdáját jelölik, melybe beleértendő a dialógus felhasználókon kívül a generikus, technikai, és privilegizált felhasználókkal elkövetett események.
- A sötétkék táblák (Információrendszer adatok, Munkafolyamat, Információrendszer környezetek, Programok, Információrendszer almodulok) a tranzakciók infrastruktúrára vonatkozó adatait és technikai információkat tartalmaznak.
- A világoskék táblák (Szervezeti egység, Szervezeti kódok) adott tranzakció szervezeti funkciójára vonatkozó információkat szolgáltatja.
- Lila fejléccel (Biztonsági besorolás) szerepel adott üzleti adattábla és audit bejegyzés biztonsági osztálya.
- A piros táblák (Engedélyezett portok, Hálózati címek) technikai adatokat írnak le a számítógépes hálózatról.

- A világosszürke adattábla (Információrendszer – környezet) kapcsolótábla az Információrendszer és Információrendszer környezetek adattáblák között.

A cél olyan adatstruktúra létrehozása, mely bármely rendszerben keletkezett naplóbejegyzés beillesztésére képes, azonban, eddigi információrendszerekre vonatkozó ismereteimre hivatkozva, a tervezett adatbázis alkalmasnak bizonyul SAP R3, Microsoft Navision, Axapta és Oracle környezetben az adatok hatékony tárolására, majd azok feldolgozására.

Amennyiben nem értelmezhető egy adott eseményre a teljes rekord feltöltése pl. felhasználói bejelentkezés (ez esetben nem történik tábla művelet), akkor az üresen hagyott változók értékkészletének bővítésére van szükség, és annak egyértelmű jelölésére, hogy adott rekordban adott attribútum nem értelmezhető, nem pedig hiányzó adat.

Az adatbázis táblái egymással szoros összeköttetésben állnak, az adatbázis konzisztencia (hivatkozási integritás) fenntartása érdekében az adattáblák tervezése szigorúan a relációs adatbázis tervezésének szabályait követi (Ullmann – Widom, 2009).

Ebben a fázisban történik a jóságmetrikák definiálása a teljeskörű objektivitást elősegítve, mivel a jóságmetrikáknak még a modellfejlesztés előtt rendszerezett struktúrában meghatározásra kell, hogy kerüljenek.

## 2.4. Változók kódolása, adattisztítás

Az adathalmaz különböző mérési skálán szereplő adatokat tartalmaz, melyek lehetnek nominális (névleges), ordinális (rendezéses), intervallum (különbség), vagy arányskálán mért adatok (Szűcs et al., 2008). A nominális és ordinális skálán mért adatok szövegszerű értékeket tartalmazhatnak, ezért ezen adatok kódolására van szükség, hogy a gépi tanuló eljárások képesek legyenek az adatfeldolgozásra (Bodon, 2010). Au (2017) a „one-hot-encoding” módszert ajánlja, melynek lényege, hogy a szöveges értéket tartalmazó függő változókból új változók létrehozásával, a változó különböző értékeinek számával megegyezően, orvosolni lehet a problémát, ahol a változó csak ott vesz fel pl. 1 értéket, ahol az az eredeti változóban adott érték jellemző volt rá. Tehát a változók kódolásának célja az összes adat számszerűsítése az eredeti teljes információtartalom megtartásának feltételével. Ehhez a ponthoz tartozik még az adattisztítás, azaz, egyrészt a hiányzó mezők felismerése és azok orvoslása, továbbá, felmerülhet a probléma, hogy az adatok exportálásának folyamatában pl. az egyes helyiértékek elcsúsznak, alapvetően numerikus értéket tartalmazó oszlopokba szöveg kerülhet stb., így a

modellfejlesztés előtt az adatbázis tételes ellenőrzése szükséges, hogy annak integritása, vagyis pontossága és teljessége, biztosított legyen. Az adatbázis felállítását, változók kódolását és adattisztítást követően az adathalmaz tulajdonosával ellenőrzési gyakorlatot szükséges tartani, hogy a feldolgozásra váró adatbázis valóban megegyezik az eredeti rendszerben tárolt adatbázissal. Ezen a ponton az adatbázist verziószámmal kell ellátni és biztonsági másolatokat kell készíteni.

## 2.5. Tréning, teszt, és validációs halmaz kiválasztása

Ahhoz, hogy megfelelően validálni lehessen az elkészítendő modell performanciáját az adatokat tréning-, teszt-, és validációshalmazra szükséges bontani (Combs, 2016). A tréning adatok szolgálnak a modell tanítására, vagyis a mintázatok felfedésére és azok elraktározására. A teszthalmazon, mely független a tréninghalmaztól, elvégzett számítások adnak becslést a modell általánosító-képességére, mivel olyan adatokat tartalmaz, melyet a modell korábban nem látott, így alkalmas a modell tesztelésére. A validációshalmaz alkalmazható a kiválasztott modellek további tuningolására, vagyis a belső paraméterek optimális kombinációjának megtalálására. A tervezett adatbázist az alábbi felbontásban ajánlott szétválasztani Raschka (2015) alapján:

- Tréninghalmaz: 60%
- Teszthalmaz: 30%
- Validációshalmaz: 10%

## 2.6. Modellfejlesztések

Miután az adatbázis megfelelő minőségben rendelkezésre áll, kezdetét veszi a modellfejlesztés, melynek legelső lépése az adatok transzformációja. Egyes modellek pl. neurális hálók, szupport vektor gépek stb., érzékenyek, ha az adatok nem ugyanazon a skálán mérhetők, mivel performanciájuk nagyban függ az alkalmazott adattranszformációs eljárástól. Az adatok transzformációja lehetséges, többek között, az adatok standardizálásával vagy normalizálásával. A döntési fa alapú algoritmusok egy tipikus példája azon eljárásoknak, melyek esetében nem szükséges az adatok transzformációja, mivel az nincs kihatásra a modell teljesítőképességére (Raschka, 2015). Az adattranszformáció legtöbb esetben az adatfeldolgozási szakaszba esik, azonban, mivel különböző modellek építése a cél, így ezt a

folyamatot a modellépítésben kell rögzíteni, mert ahogy kifejtésre is került, különböző modellek, különböző adattranszformációt igényelhetnek.

Kutató munkámban az alábbi modelleket kívánom elsősorban alkalmazni, mely összhangban áll a Disszertációs Részben felsorakoztatott modellekkel[1]:

1. **Logisztikus Regresszó (Logistic Regression).** A logisztikus regresszió egy olyan módszer, mely egy bináris output valószínűségét hivatott előrejelezni az inputváltozók alapján. Az algoritmus futása során az input változók egy súlyvektorral kerülnek összeszorzásra, melyet egy logisztikus függvény aktivál. A logisztikus regresszión alapuló gépi tanuló eljárás a súlyok folyamatos változtatásán keresztül tanul, mely az alkalmazás négyzetes hibáján keresztül realizálódik, vagyis a hiba nagysága korrekcióra kerül a logisztikus aktivációs függvény deriváltja szerint.
2. **Szupport Vektor Gépek (Support Vector Machines, avagy SVM).** Az SVM alkalmazások fő célkitűzése az egyes osztályok közötti döntési határok maximalizálása. Ha az osztályok nem szeparálhatók lineárisan, akkor az SVM kernelizálható, ami egy magasabb dimenzionális térben képezi le a mintákat. Mivel az osztályokat elválasztó szeparátor maximalizálásra kerül, ezért az SVM alkalmazások kevésbé kitettek a túlilleszkedésnek.
3. **Döntési Fa (Decision Tree).** A döntési fa alapú gépi tanuló eljárásokban a modell kérdések sorozatára válaszol, mely végső soron adott osztály meghatározásához vezet. A döntési fa az egyes attribútumok szerint kerül „szétdarabolásra” a célváltozó függvényében.
4. **Véletlen Erdő (Random Forest).** A véletlen erdők hasonlóak a döntési fák működéséhez, azonban több döntési fa általi predikciót kombinálnak, ahol a fák véletlen vektorok alapján kerülnek meghatározásra.
5. **Neurális Hálók (Neural Networks).** A neurális hálók alapkoncepciója, és amiről a nevét is kapta, az emberi agy biológiai neuronjait hivatott gépi formában megtestesíteni. Hasonlóan a természetes neuronok felépítéséhez, a mesterséges neuronok is kapcsolódnak egymáshoz, ahol egy közvetítő közeg szállítja a feldolgozott információt neuronról-neuronra. A neuronok különböző rétegekben helyezkednek el, az input réteg fogadja a feldolgozáshoz szükséges bemenő adatokat, az output réteg prezentálja a prediktált kimenetet, a közbelső rétegek, pedig transzformálják és segítségükkel jut el

[1] A modellek ugyanebben a formában közlésre kerültek a Disszertációs Részben.

az információ az input neurontól az output neuroning. A kapcsolóelem a neuronok között a dendrit, mely létezhet bármely neuron pár között, de a kapcsolat mérete gyakorta az, hogy minden neuron kapcsolódik minden másik következő rétegbeli neuronhoz. A megküldött információ az úton az egyik neurontól a másik felé átalakul a dendritek súlyaitól függően, mely végső soron az adatok tudásanyagát tartalmazza, és melyet a neurális háló a megadott tanulási eljárás és az adatokban fellelhető minta szerint módosít. A fogadó neuron a hozzá beérkező adatokat összesíti, majd egy aktivációs függvény segítségével eldönti, hogy a számára fogadott input, milyen outputként távozik.

6. **K-legközelebbi Szomszéd (K-nearest Neighbors, avagy KNN).** A KNN algoritmus egy példája a kevés paraméterrel rendelkező gépi tanuló eljárásoknak, ezért azt „lazy", azaz lusta algoritmusnak is nevezi a szakirodalom. A KNN a többségi szavazás elvén működik, azaz egy adott mintát a hozzá legközelebb álló ismert mintákhoz sorolja.
7. **Bayesian Hálók (Bayesian Network).** A Bayesian hálók olyan tanuló eljárások, melyek gráfként szemléltetik a következtetés logikáját és csomópontjai valószínűségi változók. Az algoritmusok feltételes valószínűségtáblák alapján operál.
8. **Fuzzy mintázatfelismerés (Fuzzy Pattern Recognition).** Az algoritmus a fuzzy halmazelmélet és fuzzy logika alapján működik, mely a klasszikus logikával ellentétben engedi, hogy egy állítás 0 és 1 közötti bizonyossággal igaz legyen, ezért kifinomultabb logikai következtetéseket enged.
9. **Evolúciós optimalizálás (Evolutionary Optimization).** Az evolúciós optimalizálás alapvetően kereső eljárás, melyet az evolúciós biológia inspirált. Egy adott problémára való megoldások halmazát egy kezdeti génpopulációként kezel, melyek evolúciós folyamatok mentén válnak egyre jobb megoldássá. Ez a folyamat addig iterálódik, amíg egy előre definiált fitnesszfüggvény nem lesz optimális, vagy a problémára egy elfogadható megoldás (szuboptimális válasz) születik.
10. **Klaszterező eljárások.** A klaszterező eljárások az előre nem ismert minták felderítésében szolgálnak segítségül, mely algoritmusok futásának eredményeképp az egymásra legjobban hasonlító rekordok egy közös klaszterbe kerülnek. Hátrány lehet (pl. K-közép esetén), hogy a klaszterek számát előre kell definiálni, ezért a megfelelő paraméter kikísérletezése időigényes feladat lehet.

A modellválasztásnál fontos kritérium, hogy adott modell legyen képes az input és output adatok, vagy az input és output adatok logikus transzformáció útján leképezett új változók

feldolgozására és értelmezésére. Az ismertetett algoritmusok alkalmazhatók a korábban szükségesnek ítélt adatstruktúra adatfeldolgozására.

A hibrid modellfejlesztés során célom négy különböző hibridizált modell elkészítése, melyek a diagramon ábrázolt kutatói munkamenetben (1. melléklet) az alábbi címkéket viselik:

- Hibrid Modell I.
- Hibrid Modell II.
- Hibrid Modell III.
- Nem Hibrid Modell (azaz 0. szintű hibrid modell)

A hibrid modellezés legelső fázisa a hibridizáció fokának objektív levezetése. Az elkészítendő modellekről matematikailag igazolhatóan el kell tudni dönteni, hogy melyik bír magasabb fokú hibrid elemekkel. Így a legelső modell tekinthető a „leghibridebbnek", míg a negyedik modell a legkevésbé hibrid eljárás. H1 hipotézisemmel összhangban, a célom annak bizonyítása, hogy a hibridizáció növelésével magasabb teljesítmény érhető el, így négy modellen elvégezve a tudományos szimuláció matematikailag kiértékelhető.

A modellfejlesztés során történik a modell tanítása, mely folyamatban a tréningadat kerül a modellnek betöltésre, mely végig iterál az összes rekordon és matematikailag leképezi a minták közötti összefüggéseket súlyvektorokba (kivéve a KNN és Döntési fa eljárásoknál). Amennyiben alacsony performanciát produkál a modell, szükséges annak paramétereinek optimalizálása, melyre vonatkozóan empirikus kísérleteket szükséges elvégezni. Mivel a paraméterek száma és fajtája modellenként eltérő, ezért általánosságban az alábbi optimalizálási eljárásokat érdemes figyelembe venni Baesens et al. (2015) és Fawcett és Provost (2013) alapján[2]:

1. **Túlilleszkedés csökkentése.** Túlilleszkedés akkor merül fel, ha a modell alaposan megjegyezte a tréninghalmazban meghúzódó mintákat vagy véletlen zajokat, azonban a teszthalmazon alulteljesít. A problémát a modell komplexitásának csökkentésével pl. függvény regularizációval lehet kivitelezni.
2. **Tanulási ráta növelése/csökkentése.** A tanulási ráta a modell konvergenciáját hivatott lassítani vagy gyorsítani. Magas tanulási ráta esetén a konvergencia gyorsabb, azonban így könnyebben átugorhatók a hibafüggvény minimumpontjai, míg alacsony tanulási ráta esetén a konvergencia lassabb, azaz a tanulás költségesebb, ellenben, garantált a

[2] Az eljárások ugyanebben a formában közlésre kerültek a Disszertációs Részben.

hibafüggvény egy lokális minimumának megtalálása. Fontos kiemelni, egyik esetben sem biztosított a globális minimumpont felfedezése, ezért a gépi tanuló eljárások súlyvektorainak véletlen változtatása javasolt.

3. **Tréningelés növelése.** Könnyen meglehet, hogy a tréninghalmaz egyszeri „megmutatása” a gépi tanuló eljárásnak nem jár sikerrel, vagyis nem képes a mintázatot teljes egészében leképezni, ezért a tréningelés számának növelése szükséges lehet a processzoridő kárára.
4. **Aktivációs függvény változtatása.** A klasszifikációs gépi tanuló algoritmusok egy része aktivációs függvényt alkalmaz a klasszifikáció valószínűségének meghatározására pl. szigmoid, tangens, RELU stb. Egy adott aktivációs függvény egy adott problémára nem biztos, hogy optimális eredményt tud adni, ezért empirikus úton meggondolandó kísérletezni különböző aktivációs függvényekkel.
5. **Architektúra változtatása.** Amennyiben az egyes modellparaméterek optimalizálása során is a modell alulteljesít, indokolt lehet a teljes architektúrát újragondolni.

A paraméterek optimalizálását követően történik a visszaellenőrzés és megszületnek a végső kimenetek, azaz a modellek. A gépi tanuló modellek egyik legnagyobb hátránya, hogy nem lehetséges általuk hipotézisek és konfidencia intervallumok tesztelése, tehát hiányzik belőlük a statisztikai módszerek alap karakterisztikái. Az ötödik elkészítendő modell egy tradicionális statisztikai modell lesz, ezért a fejlesztés menete némileg eltér a gépi tanuló rendszerek fejlesztésétől.

## 2.7. Eredmények összehasonlítása I.

A modellalkotást követően azok kiértékelését kell elvégezni annak érdekében, hogy a teljesítményt mutatószámokkal ki lehessen fejezni. A jóság kritériumokra előzetesen metrikákat kell építeni, hogy objektivitása biztosítható legyen. Az anomália észlelés során az alábbi jóság metrikákat és kimutatásokat kívánom alkalmazni és elemezni[3]:

- **Relatív hiba.** A becslés abszolút hibáját adja meg a megfigyelt értékek tükrében.
- **Helytelen becslések aránya.** A modell helyes becslésének valószínűsége.

[3] A metrikák ugyanebben a formában közlésre kerültek a Disszertációs Részben.

- **Álpozitív becslések aránya.** Az álpozitív találatok aránya. Az anomália észlelés során az egyik legfontosabb mutató, mely azt az információt szolgáltatja, hogy az igazolt anomáliákat milyen arányba fogadta el a rendszer normális magatartásként.
- **Álnegatív becslések aránya.** Az álnegatív találatok aránya.
- **Átlagos négyzetes hiba.** A kimenet négyzetes hibaösszege.
- **Vevő Működési Karakterisztika (Received Operating Characteristic).** A klasszifikáció álpozitív és álnegatív aránya közötti kompromisszum grafikus megjelenítése.
- **Validációs görbe.** Az egyes adathalmazok (tréning, teszt és validációs) tanulásának pontosságát hivatott bemutatni a minta időbeli feldolgozásának tükrében. Ha a tréninghalmaz és teszt-, és validációshalmaz görbéi közötti különbség jelentős, akkor a rendszer nagyvalószínűséggel túlilleszkedett, azaz megtanulta az adathalmazban jelentkező véletlen zajokat, vagy „memorizálta" a tréninghalmazt.
- **Igazságmátrix.** Az igazságmátrix grafikusan szemlélteti az igaz pozitív és igaz negatív, valamint az álpozitív és álnegatív értékeket.
- **F1-pontszám.** Az F1-pontszám egy 0 és 1 közötti érték, mely kétszer a modell pontossága és igaz pozitív mutatójának szorzata és a modell pontossága és igaz pozitív mutatójának hányadosa.
- **Tanulás gyorsasága.** A tanulás gyorsasága azt szemlélteti, hogy a modellt hányszor kellett lefuttatni a tréninghalmazon, amíg azt elfogadható szinten megtanulta.
- **Keresztvalidáció.** A keresztvalidáció a modellek pontosságát mutatja különböző validációshalmazok véletlen kiosztásával.

Fontos azonban kiemelni, hogy a jóságmetrikák kiértékelése egy folyamatos tevékenység, a paraméterek optimalizálása előtt, után és közben is szükséges azok mérése, hogy igazolhatóvá váljon, hogy a paraméter-optimalizálási eljárás pozitív irányba halad.

Az eredmények összehasonlítása adja alapját a H1 és H2 hipotézis bizonyításának, vagy elvetésének.

## 2.8. Eredmények összehasonlítása II.

Az eredmények összehasonlítása második fázisában történik a statisztikai modell és a hibrid modellek közötti eredmények összehasonlítása a H3 bizonyítása vagy elvetése a felállított jóságmetrikák alapján.

## 2.9. Eredmények összehasonlítása III.

Az eredmények összehasonlításának harmadik fázisában történik a védelmi mechanizmusok és adattranszformáció lehetséges vizsgálata a külső manipulációkkal szembeni sérülékenységek feltárása. Az adatok olyan szintű transzformációjára van szükség, melyek változtatás esetén is képesek az eredeti adat visszaállítására, vagy a változtatások zajszintű kiszűrésére.

## 2.10. Következtetés

A kutatómunka következtetési szakasza fogja összefoglalni az elvégzett munka és eredmények konklúzióját, melyben kifejtésre kerül, hogy a felállított hipotézisek elfogadásra, vagy elvetésre kerülnek.

# 3. Összefoglalás

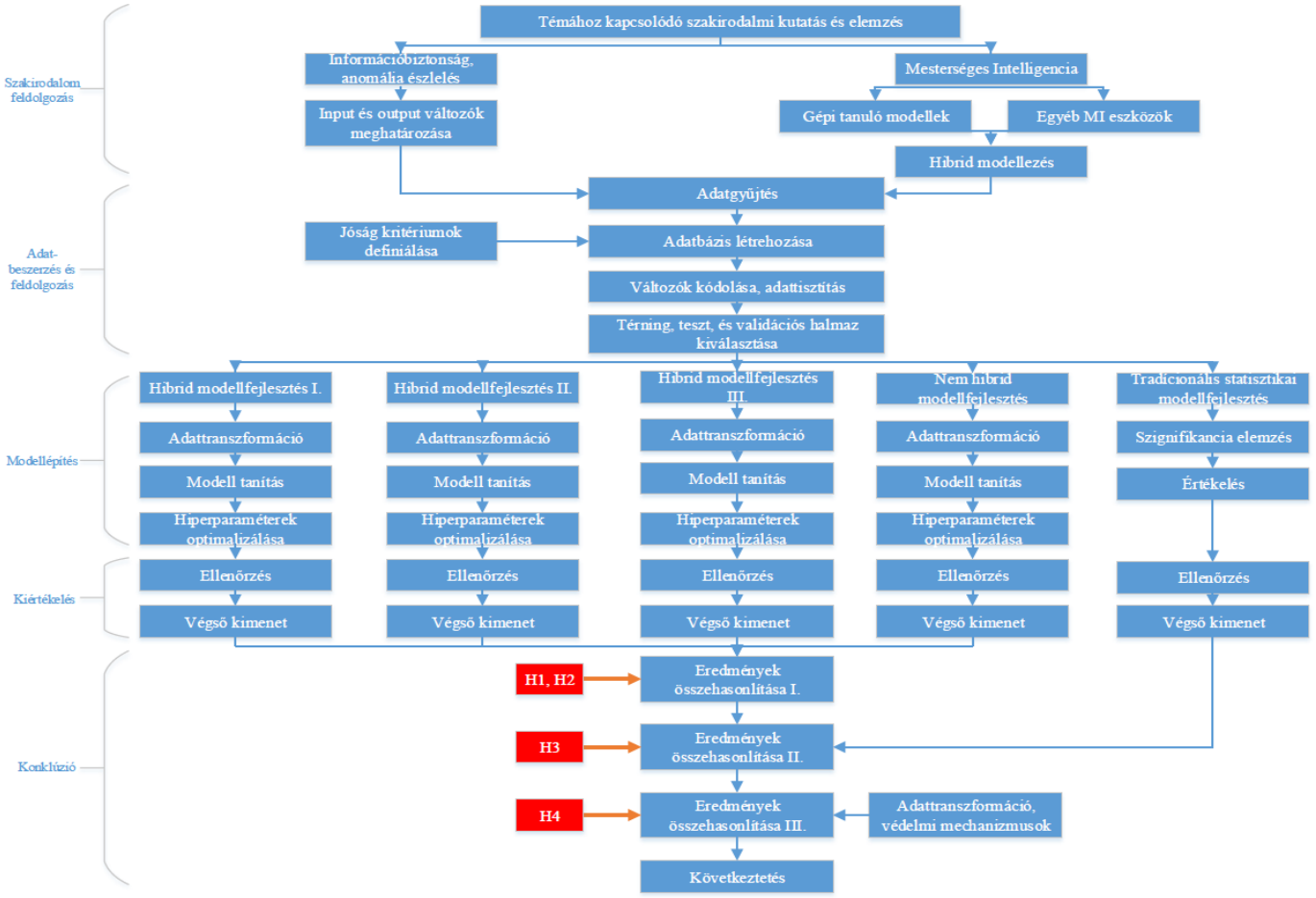
A dokumentumban kifejtésre került a következő periódus tervezett kutatási programja és munkaterve. A kutatási legelső lépése a témához kapcsolódó szakirodalmi kutatás és elemzés, mely alapján a szükségesnek ítélt változók és alkalmazni kívánt módszerek meghatározásra kerülnek. A következő lépés a naplóállományokra vonatkozó adatgyűjtés, melyet primer kutatás keretében különböző szervezetektől kívánok beszerezni. Az adatgyűjtést követően felállítom az adatbázist a 2. számú mellékletben bemutatott struktúra szerint, majd implementálom adatfeldolgozásra a kutatásra szánt belső környezetre. Ezúton a változók kerülnek kódolásra, majd az adathalmaz felosztása tréning-, teszt-, és validációs halmazokra. Miután az adatok előfeldolgozása megfelelő, kezdetét veszi a modellezés, a tervezett 5 modell előállítása. A modellfejlesztés után az eredményeket összehasonlítom és leszűröm a konklúziót, mely a felállított hipotéziseket megerősítik, avagy cáfolják.

# 4. Irodalomjegyzék


1. Au T. C. (2017): Random Forests, Decision Trees, and Categorical Predictors: The „Absent Levels" Problem. https://arxiv.org/pdf/1706.03492.pdf. Letöltés ideje: 2018. május 9.
2. Baesens B. – Vlasselaer V. V. – Verbeke Q. (2015): Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection. USA, Wiley and SAS Business Series, 359 p.
3. Banko M. – Brill E. (2001): Scaling to Very Very Large Corpora for Natural Language Disambiguation. http://www.aclweb.org/anthology/P01-1005. Letöltés ideje: 2018. május 19.
4. Bodon F. (2010): Adatbányászati algoritmusok. Budapest, Free Software Foundation, 278 p.
5. Combs A. (2016): Python Machine Learning Blueprints: Intuitive data projects you can relate to. 1st Edition, Kindle Edition. Birmingham, Pack Publishing, 332 p.
6. Fawcett T. – Provost F. (2013): Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. USA, O'Reilly Media, 414 p.
7. Raschka S. (2015): Python Machine Learning. Birmingham, Packt Publishing, 425 p.
8. Szűcs I. – Lőkös K. – Obádovics Cs. – Szigetváriné J. É. Zs. – Bárdos I. K. – Széles Zs. – Késmárky-Gally Sz. – Mucsics L. – Mohamed Zs. (2008): A tudományos megismerés rendszertana. Budapest, Szent István Egyetem Kiadó, 272 p.
9. Ullmann J. D. – Widom J. (2009): Adatbázisrendszerek. Alapvetés. Budapest, Panem, 600. p.

# 5. Mellékletek

## 1. számú melléklet: Kutatási munkaterv-diagram



## 2. számú melléklet: Tervezett adatbázis-struktúra