

ДОДАТОК А

Програмні коди

1) Реалізація CBR-цикла

```

private void btnSearch_Click(object sender, EventArgs e)
{
    var _rangesDmax = _db.RangesDMax(CaseName);
    var lstResultCases = new List<ResultCase>();
    predMax = 0;
    foreach (var rd in _rangesDmax)
    {
        predMax = rd + predMax;
    }
    d_Max = Math.Sqrt(predMax);
    DataTable selTestData = new DataTable();
    selTestData = (DataTable)dgvTestCases.DataSource;
    if (dgvTestCases.SelectedRows.Count <= 0)
    {
        MessageBox.Show("Select test value");
        return;
    }
    int selectedRowIndex = dgvTestCases.SelectedRows[0].Index;
    var tmp =
    GetSelectedTestCaseValue(selTestData,
    selectedRowIndex); for (int i = 0; i <
    dgvCases.RowCount; i++)
    {
        double sub = 0;
        for (int j = 0; j < dgvCases.ColumnCount; j++)
        {
            if (dgvCases.Columns[j].HeaderText == "Result" ||
            dgvCases.Columns[j].HeaderText == "Name")
            {
                continue;
            }
            if (tmp[j].Typ == typeof(System.Double) || tmp[j].Typ ==
            typeof(System.Int32))
            {
                sub += double.Parse(dgvCases[j,
                i].Value.ToString()).DCjitSingle(double.Parse(tmp[j].V
                al)); continue;
            }
            if (tmp[j].Typ == typeof(System.String))
            {
                sub += dgvCases[j,
                i].Value.ToString().String_DCjitSingle(tmp[
                j].Val); var mm = dgvCases[j,

```

```

        i].Value.ToString().String_DCjitSingle(tmp[
        j].Val); continue;
    }
    if (tmp[j].Typ == typeof(System.Boolean))
    {
        sub += bool.Parse(dgvCases[j,
        i].Value.ToString()).Bool_DCjitSingle(bool.Parse(tmp[j
        ].Val)); continue;
    }
}
double Distant_i = Math.Sqrt(sub);
double Delta_I = (1 - (Distant_i / d_Max)) * 100;

double _k = double.Parse(nudK.Text);
if (Delta_I > _k)
{
    ResultCase rc = new ResultCase();
    rc.Result = dgvCases["Result", i].Value.ToString();
    rc.Name = dgvCases["Name", i].Value.ToString();
    rc.Similarity = Math.Round(Delta_I, 2);
    var lstKey = lstResultCases.Where(x => x.Name
    == rc.Name).ToList(); if (lstKey.Count <= 0)
    {
        lstResultCases.Add(rc);
    }
}
}
if (lstResultCases.Count > 0)
{
    for (int i = 0; i <= lstResultCases.Count-1; i++)
    {
        string r = lstResultCases[i].Result;
        int countk = 0;
        for (int j = 0; j <= lstResultCases.Count-1; j++)
        {
            if (r == lstResultCases[j].Result)
            {
                countk = countk + 1;
            }
        }
        if (countk < int.Parse(textBox1.Text))
        {
            for (int j = 0; j <= lstResultCases.Count-1; j++)
            {
                if (r == lstResultCases[j].Result)
                {
                    lstResultCases.RemoveAt(j);
                    j = j - 1;
                }
            }
        }
    }
}

```

```

        }
    }
    i = i - 1;
}
}
}

```

Процедура виділення центроїдів

```

private void btnCentroid_Click(object sender, EventArgs e)
{
    frmCase f = new frmCase();
    string a = "";
    string k = f.Centroid(ref a);
    adp = _db.Adapter(k);
    ds.Clear();
    adp.Fill(ds, CaseName);
    dgvCases.DataSource = ds.Tables[CaseName];
}
public string Centroid(ref string k)
{
    k = String.Format("select Round(AVG(STG),2) as
    STG, Round(AVG(SCG),2) as SCG, Round(AVG(STR),2) as
    STR, Round(AVG(LPR),2) as LPR, Round(AVG(PEG),2) as PEG, Result from {0}
    Group by Result", CaseName);
    return k;
}
}

```

Процедура обчислення ступкню подібності

```

private double CheckQuality()
{
    var _db = new AppDB();
    var BP = _db.GetTable(CaseName);
    var TD = _db.GetTable("CaseTestData");
    var Ranges = _db.RangesDMax(CaseName);
    var M = 0;
    var GivenID = "";
    for (int i = 0; i < TD.Rows.Count; i++)
    {
        DataRow TDRow = TD.Rows[i];
        var lstResultCases = new List<ResultCase>();
        double d_Max = 0;
        double predMax = 0;
        foreach (var rd in Ranges)

```

```

{
    predMax = rd + predMax;
}
d_Max = Math.Sqrt(predMax);
for (int h = 0; h < BP.Rows.Count; h++)
{
    DataRow BpRow = BP.Rows[h];
    double sub = 0;
    string caseId = "";
    string caseKey = "";
    for (int k = 0; k < BP.Columns.Count; k++)
    {
        if (BP.Columns[k].Caption == "Name")
        {
            caseKey = BpRow["Name"].ToString();
            continue;
        }
        if (BP.Columns[k].Caption == "Result")
        {
            caseId = BpRow["Result"].ToString();
            continue;
        }
        if (TD.Columns[k].DataType == typeof(System.Double)
|| TD.Columns[k].DataType == typeof(System.Int32))
        {
            sub +=
                double.Parse(BpRow[k].ToString()).DCjtSingle(double.Par
                se(TDRow[k].ToString())); continue;
        }
        if (TD.Columns[k].DataType == typeof(System.String))

        {
            sub +=
                BpRow[k].ToString().String_DCjitSingle(TDRow[
                k].ToString()); continue;
        }
        if (TD.Columns[k].DataType == typeof(System.Boolean))
        {
            sub +=
                bool.Parse(BpRow[k].ToString().ToString()).Bool_DCjitSingle(bool.Pars
                e(TDRow[k].ToString()));
            continue;
        }
    }
    double Distant_i = Math.Sqrt(sub);
    double Delta_I = (1 - (Distant_i / d_Max)) * 100;
    ResultCase rc = new ResultCase();
    rc.Result = caseId;
    rc.Name = caseKey;
}

```

```

    rc.Similarity = Delta_I;
    var lstKey = lstResultCases.Where(x => x.Name ==
rc.Name).ToList();
    if (lstKey.Count <= 0)
    {
        lstResultCases.Add(rc);
    }
}
if (lstResultCases.Count > 0)
{
    var tmp = lstResultCases.OrderByDescending(x =>
x.Similarity).Select(x => x.Result).FirstOrDefault(); GivenID
= tmp;
    bool flag = GivenID.Equals(TDRow["Result"].ToString());
    if (flag)
    {
        M = M + 1;
    }
    var nc = new NewCaseValue()
    {
        PKey = TDRow["Name"].ToString(),
        OldResult = int.Parse(TDRow["Result"].ToString()),
        NewResult = int.Parse(GivenID)
    };
    NewResult.Add(nc);
}
}
double percentage = 0;
percentage = ((double)M / TD.Rows.Count) * 100;
return percentage;
}
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i <= dgvResultcases.RowCount-1; i++)
    {

        string st =
dgvResultcases.Rows[i].Cells[0].Value.To
String(); bool fl = true;
        for (int j = 0; j < dataGridView1.RowCount-1; j++)
        {
            string val = "";
            if (dataGridView1.RowCount > 1)
            {
                val = dataGridView1.Rows[j].Cells[0].Value.ToString();
            }
        }
    }
}

```

```
    }
    if (val == st)
    {
        fl = false;
    }
}
if (fl == true)
{
    int countk = 0;
    double simk = 0;
    for (int j = 0; j <= dgvResultcases.RowCount - 1; j++)
    {
        string st1 =
            dgvResultcases.Rows[j].Cells[0].Value.ToString();
        if (st1 == st)
        {
            countk = countk + 1;
            simk = simk +
                double.Parse(dgvResultcases.Rows[j].Cells[2].Value.ToString
                    ());
        }
    }
    simk = simk / countk;
    dataGridView1.Rows.Add(st, simk);
}
}
dataGridView1.Sort(dataGridView1.Columns[1],
    ListSortDirection.Descending);
}
```

ДОДАТОК Б

Слайди презентації

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

Кафедра ПІ

Атестаційна робота
магістра

**Дослідження методів витягу нових знань в
системах підтримки прийняття рішень**

Виконав
ст.гр. ПЗСзм-18-1
Фареник І.В.

Науковий керівник
проф. Шостак І.В.

1 1

Мета роботи

Метою роботи є дослідження й розробка методів і програмних засобів ІАД на основі прецедентів для СКБД і СКБЗ.

Запропоновано модифікований CBR метод (CBR цикл), що використовує експертну інформацію (тестові набори даних) для витягу прецедентів, який підвищує якість рішення завдань ІАД на основі прецедентів за рахунок формування бази вдалих (відповідних) і невдалих (невідповідних) прецедентів у процесі виконання CBR циклу

1 2

- Описано алгоритми витягу прецедентів і розроблений модифікований алгоритм витягу прецедентів на основі k - Nn для ІАД, що полягає в зміні значення k залежно від розміру БП.
- Модифікація дозволяє підвищити якість рішення завдань ІАД, зокрема, підвищити якість класифікації даних з використанням СВР методу при збільшенні розміру БП.

3

Процес виявлення знань у БД

це процес пошуку корисних знань у сирих даних

- вибір предметної області й релевантного знання для реалізації цілей кінцевого користувача комп'ютерної системи;
- вибір вихідної множини даних і підмножини змінних, які необхідні для витягу нового знання з бази фактів;
- уточнення даних і предпроцесінг: вибір основних операцій над даними так, що вони можуть сприяти зменшенню «шумів», визначення стратегій для їхньої мінімізації;
- редукція даних: виявлення корисних особливостей даних, щоб представлення даних було адекватним рішенням завдань, відповідних до поставлених цілей;
- вибір завдання DM, тобто специфікація процесу як класифікації, кластеризації і т.д.;
- вибір алгоритмів, що реалізують DM для пошуку зразків (patterns) у даних. Цей вибір повинен бути погоджений з моделями й параметрами представлення даних;
- DM: пошук зразків у формі, зручної для користувача (правила класифікації й кластеризації, регресія, дерева рішень і т.д.);
- інтерпретація породжених зразків з можливим повторенням етапів 1-7 для подальшої ітерації. Цей етап часто має на увазі використання методів, що перебувають на стику технології DM і технології експертних систем (ЕС). Від того, наскільки ефективним він буде, у значній мірі залежить успіх рішення поставленого завдання;
- огляд і узгодження виявленого знання.

4

Рішення завдань прогнозування розбивається на два етапи

- На першому етапі на підставі набору даних з відомими результатами будується модель.
- На другому етапі вона використовується для проорокування результатів на підставі нових наборів даних.
- При цьому, потрібно, щоб побудовані моделі працювали максимально точно.
- До даного виду завдань відносять завдання класифікації й регресії, а також пошук асоціативних правил, якщо отримані правила можуть бути використані для проорокування появи деяких подій.

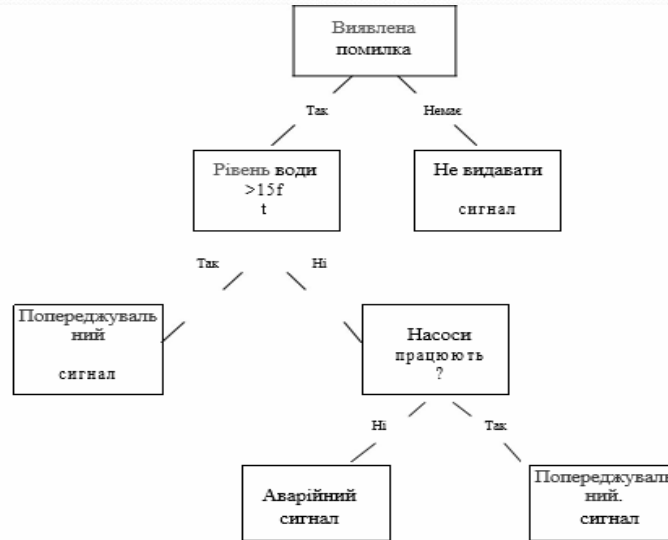
5

Завдання ІАД

- класифікація;
- кластеризація (сегментація);
- регресія;
- прогнозування;
- пошук асоціативних правил;
- аналіз послідовностей;
- аналіз відхилень;
- оцінювання;
- аналіз зв'язків;
- візуалізація даних і ін.

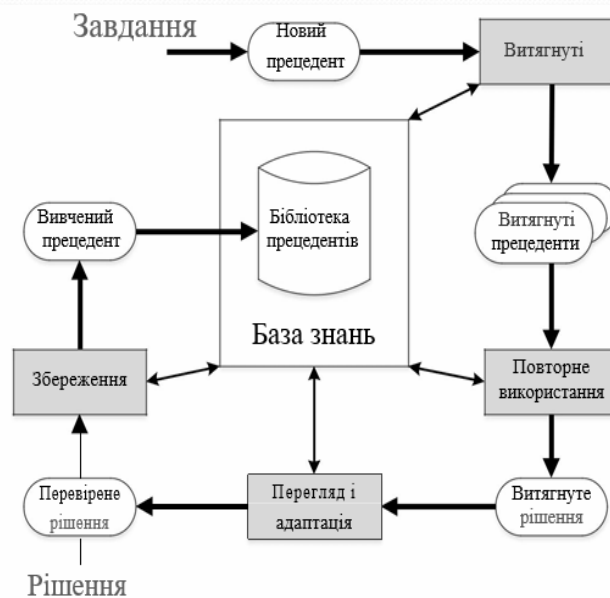
6

Дерево рішень класифікації несправних станів технічного об'єкта



7

Структура СВР циклу

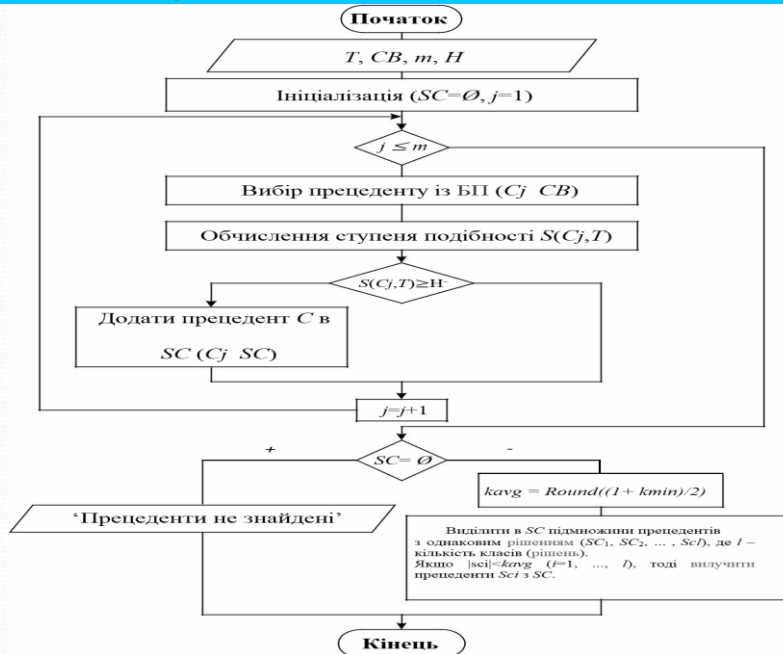


8



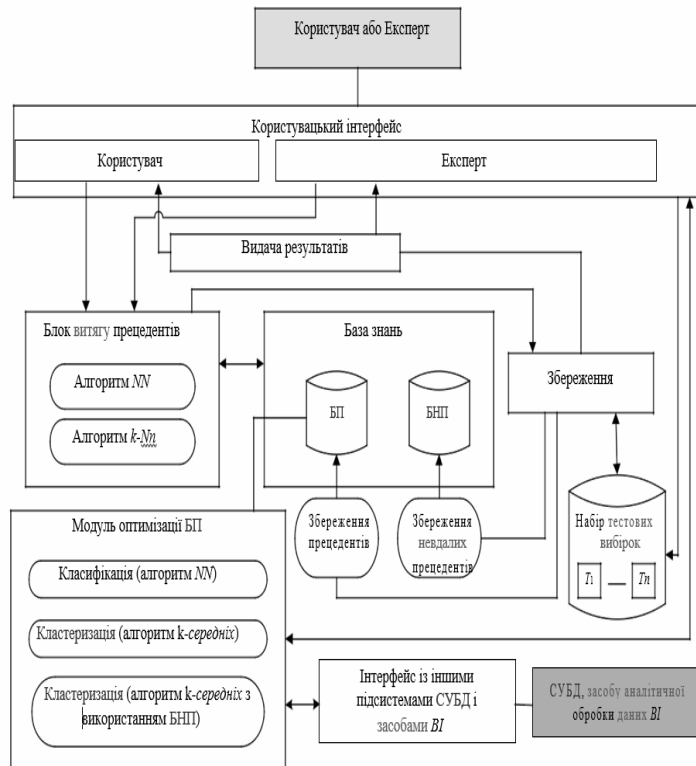
11

Схема модифікованого алгоритму витягу прецедентів на основі k - Nn



12

Архітекту ра прототипу СВР системи



13

Діалогове вікно для завантаження початкової БП

14

Діалогове вікно для введення значень параметрів поточної ситуації

Рассуждение на основе прецедентов

Метрика для обчислення ступеня збіжності поточної ситуації і прецедентів з БП ⇒ Метрика: Евклідова метрика

Кількість класифікацій на тестовій вибірці для даної БП ⇒ Якість класифікації: 72%

Name	STG	SCG	STR	LPR	PEG	Result
P001	0	0	0	0	0	1
P002	0.08	0.08	0.1	0.24	0.9	4
P003	0.06	0.06	0.05	0.25	0.33	2
P004	0.1	0.1	0.15	0.65	0.3	3
P005	0.08	0.08	0.08	0.98	0.24	2
P006	0.09	0.15	0.4	0.1	0.66	3

Кількість нових ситуацій: 1
Значення для алгоритму k-NN: K: 2
Порогове значення ступеня схожості: Н: 80

Значення параметрів поточної ситуації

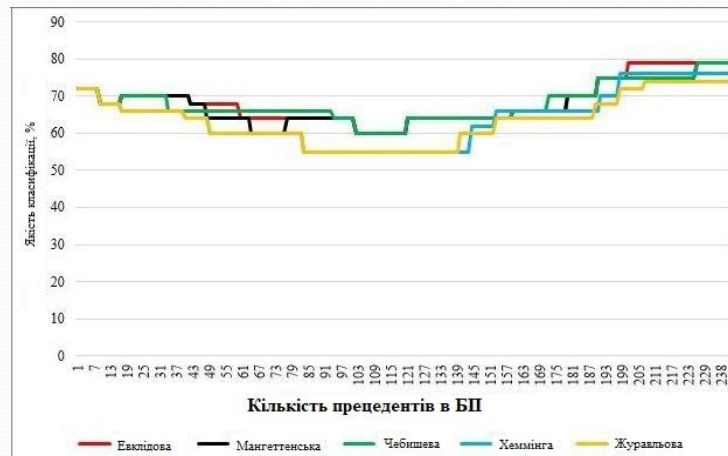
STG	SCG	STR	LPR	PEG
0.24	0.1	0.25	0.06	0.9

Назад Вихід

- Таким чином, для даного прикладу CBR система додасть новий прецедент у БП, класифікувавши дану ситуацію (рівень знань, суб'єкта навчання) як «4» – високий (high). Зверніть увагу, при додаванні нового прецеденту зростає якість класифікації з 72% до 74%.

15

Графік залежності якості класифікації від кількості прецедентів у БП із використанням різних метрик



16

Висновки

- В атестаційній роботі магістра проведено дослідження різних технологій, методів і програмних засобів ІАД, що включаються до складу сучасних СКБД, і встановлено, що однією з перспективних можливостей розширення засобів ІАД і аналітичних інструментів СКБД є використання прецедентного підходу.
- Розроблена модифікація алгоритму витягу прецедентів на основі k - Nn для ІАД, що полягає в зміні значення k залежно від розміру БП. Дана модифікація дозволяє підвищити якість рішення завдань ІАД, зокрема, підвищити якість класифікації даних з використанням CBR методу.
- Виконана програмна реалізація прототипу CBR системи для розширення можливостей засобів ІАД у СКБД на прикладі Microsoft SQL Server з використанням мови C# і середовища програмування MS Visual Studio, а також технології Windows Forms, ADO.NET Entity Framework, аналітичної платформи Deductor і SQL Server Analysis Services.
- Розглянуто приклад використання розробленого прототипу системи для рішення завдання класифікації даних з репозиторію UCI Machine Learning Repository.

ДОДАТОК В
Відгук і рецензії

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ
 ННЦ ЗФН

ВІДЗИВ
 на атестаційну роботу магістра

Фареника Іллі Вадимовича, гр. ПЗСзм 18-1

спеціальність 121 – *Інженерія програмного забезпечення*
 освітньо-професійна програма - «**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМ**»
 Тема атестаційної роботи «Дослідження методів витягу нових знань в
 системах підтримки прийняття рішень»

Методи ІАД і, зокрема, технології Data Mining сьогодні широко застосовуються в СКБД для рішення актуального завдання виявлення в даних раніше невідомих і практично корисних знань, необхідних для прийняття рішень у різних сферах людської діяльності. Знайдені закономірності повинні бути логічно з'ясовні, а якщо ні, то існує ймовірність, що вони є випадковими. Крім того, виявлені знання повинні бути представлені в зрозумілому для людини виді.

В результаті роботи розроблена імітаційна програма, що ілюструє прийняті проектні рішення і дозволяє проводити програмні експерименти з високою надійністю отриманих результатів.

Остаточний висновок щодо оригінальності роботи, враховуючи критерії оцінювання плагіату (п.4 Положення про протидію академічному плагіату в ХНУРЕ) – згідно звіту системи «Unicheck» ID перевірки: 1000752755 – робота є оригінальною (79% оригінального тексту).

Магістрант гр. ПЗСзм-18-1 *Фареник Ілля Вадимович* готовий до самостійної інженерної діяльності. Атестаційну роботу можна подати до захисту в ЕК за спеціальністю 121 – *«Інженерія програмного забезпечення»*, освітньо-професійною програмою *Програмне забезпечення систем*.

«14» 12 2019р.

Керівник атестаційної роботи магістра

І.В. Шостак, д.т.н., професор кафедри ПІ



Рецензія

на атестаційну роботу магістра
магістранта групи ПЗСзм-18-1 Фареника Іллі Вадимовича
спеціальність – 121- Інженерія програмного забезпечення
освітньо-професійна програма Програмне забезпечення систем

«Дослідження методів витягу нових знань в системах підтримки прийняття рішень»

Структура атестаційної роботи: пояснювальна записка 88 стор.;
графічна частина 17 аркушів; програмне застосування (прикладна програма)
12 файлів загальним обсягом 1200 Кбайт.

Прогнозуючі методи використовують значення одних змінних для прогнозування невідомих (пропущених) або майбутніх значень інших (цільових) змінних. До даної групи методів відносять нейронні мережі, дерева рішень, лінійну регресію, метод найближчого сусіда, метод опорних векторів і ін. У комбінації з іншими методами ІАД прогнозування припускає аналіз тенденцій, класифікацію, зіставлення з моделлю й відносини.

Зміст роботи відповідає завданню та темі атестаційної роботи. Обсяг записки та домірність окремих розділів відповідає вимогам до атестаційної роботи магістранта.

В атестаційній роботі оброблено досить велику кількість наукового матеріалу, на високому теоретичному та методологічному рівні проведено дослідження особливостей і методів обробки та покращення обчислень при виділенні та побудові границь територій.

Пояснювальна записка виконана у відповідності до вимог нормативних документів і ДСТУ.

До недоліків можна віднести те, що в роботі недостатньо уваги приділено питанням обґрунтування вибору інструментальних засобів для створеного програмного додатку.

Атестаційна робота магістранта групи ПЗСзм-18-1 Фареника Іллі Вадимовича відповідає вимогам до атестаційних робіт і заслуговує оцінки «добре 80».

Атестаційну роботу можна представити для захисту в ЕК за спеціальністю 121 – Інженерія програмного забезпечення, освітньо-професійною програмою Програмне забезпечення систем.

Рецензент,

к.т.н, доц. каф. ШІ

В.Л. Шергін

« 15 » 12 2019 р.



Рецензія

на атестаційну роботу магістра
магістранта групи ПЗСзм-18-1 Фареника Іллі Вадимовича
спеціальність – 121- Інженерія програмного забезпечення
освітньо-професійна програма Програмне забезпечення систем

«Дослідження методів витягу нових знань в системах підтримки прийняття рішень»

Структура атестаційної роботи: пояснювальна записка 88 стор.;
графічна частина 17 аркушів; програмне застосування (прикладна програма)
12 файлів загальним обсягом 1200 Кбайт.

В системах інтелектуального аналізу даних для витягу нових знань із наявних даних застосовуються різні методи: статистичні й індуктивні процедури (дерева рішень), генетичні алгоритми, штучні нейронні мережі, кластерний аналіз, прецедентні методи й ін., у роботі для виконання ІАД пропонується використовувати методи правдоподібних міркувань на основі прецедентів (СВР – Case-Based Reasoning). Такий підхід є актуальним та мало вивченим. Метою роботи є дослідження й розробка методів і програмних засобів ІАД на основі прецедентів для СКБД і СКБЗ..

Зміст роботи відповідає завданню та темі атестаційної роботи. Обсяг записки та домірність окремих розділів відповідає вимогам до атестаційної роботи магістранта.

В атестаційній роботі оброблено досить велику кількість наукового матеріалу, на високому теоретичному та методологічному рівні проведено дослідження особливостей і методів обробки та покращення обчислень при виділенні та побудові границь територій.

Пояснювальну записку до атестаційної роботи викладено з дотриманням внутрішньої логіки, між розділами простежується взаємозв'язок.

При виконанні роботи використалися методи прикладного системного аналізу, елементи теорії ймовірностей, теорія надійності програмного забезпечення, методологія мультиверсійного проектування стійкого до відмов програмного забезпечення, методологія розробки програмного забезпечення.

Недоліком роботи є недостатньо докладний опис дослідної експлуатації розроблених програмних модулів..

Атестаційна робота магістранта групи ПЗСзм-18-1 Фареника Іллі Вадимовича відповідає вимогам до атестаційних робіт і заслуговує оцінки «добре 80». Атестаційну роботу можна представити для захисту в ЕК за спеціальністю 121 – Інженерія програмного забезпечення, освітньо-професійною програмою Програмне забезпечення систем.

Рецензент,

к.т.н, проф. каф. ПІ



І.Ю. Шубін