

**А. М. Боровий, асп.**

## **ОЦІНКА ЕНЕРГОСПОЖИВАННЯ ІНСТРУКЦІЙ ОПРАЦЮВАННЯ ДАНИХ**

*Використання вбудованих та автономних систем набуває у наш час широкого поширення. Ключові вимоги до простоти швидкодії відтісняються сьогодні на задній план проблемою мінімального енергоспоживання. Оптимізацію енергоспоживання можна проводити в кількох напрямках. Найбільш універсальним з них є метод оптимізації програмного забезпечення. Для оцінки енергоспоживання програмного забезпечення рекомендовано узагальнити дані експериментів за допомогою нейронних мереж. Розглянуто методику формування навчальної вибірки для оцінки енергоспоживання на прикладі інструкцій опрацювання даних процесорного ядра ARM7TDMI.*

### **1. Вступ**

Використання вбудованих та автономних систем на сьогоднішній день отримало широке застосування. Таке застосування пояснюється наявністю відповідної елементної бази для таких пристроїв на основі КМОН-технології. Однак, якщо в попередні роки основна увага зосереджувалась на збільшенні швидкодії автономних та вбудованих систем, то в останнє десятиліття набуває значного поширення вимога збільшення тривалості роботи таких систем від автономних джерел живлення. Всі напрямки збільшення тривалості роботи автономних систем можна розділити на три групи: за рахунок збільшення ємності джерел живлення; за рахунок оптимізації внутрішньої структури компонентів; за рахунок оптимізації програмного забезпечення. Недоліки та переваги кожного з цих методів було розглянуто у [1], де зроблено висновок про перспективність третього напрямку, як найбільш універсального — він годиться як для нових, так і для існуючих систем. Цей шлях базується на математичних моделях енергоспоживання для виконання окремих інструкцій, а також переходів між ними. Однак, існуючі моделі енергоспоживання мікропроцесорів і мікроконтролерів мають невисоку точність через необхідність нехтування деякими параметрами інструкції [2] запропоновано прогнозувати енергію виконання інструкцій та переходів між ними за допомогою нейронних мереж. Найважливішим етапом реалізації такої методики є формування навчальної вибірки для прогнозуючої нейронної мережі. Проведені дослідження у [3] показали перспективність використання для цього регресійного аналізу. Метою роботи є розвиток методу формування навчальної вибірки нейронної мережі для оцінки енергоспоживання інструкцій опрацювання даних за допомогою регресійного аналізу. Як приклад використано результати вимірювання енергії процесорного ядра ARM7TDMI [4].

### **2. Методи сортування даних**

В процесі проведення досліджень, з метою мінімізації обчислень було апробовано різні методи оцінки даних. Приймаючи до уваги той факт, що інформація про методику вимірювання енергоспоживання [4] є неповною, оцінка та створення відповідних моделей є досить складною. На основі попередніх дослідів, було прийнято рішення про доцільність як попереднього сортування даних за максимальною кількістю інформації (максимальною кількістю бітів, які використовуються при формуванні, обробці та виконанні інструкції) так і про поділ інструкцій в середині кожної групи на підгрупи, що дозволить об'єднати їх на основі енергоспоживання у групи, а в кінцевому результаті призведе до зменшення кількості експериментальних досліджень.

Для спрощення операцій розрахунку у [4] було введено поняття «базового» енергоспоживання для кожного режиму адресачії. Під цим приймалося енергоспоживання інструкції, яка виконувалась при умові, що всі значення є мінімальними (дорівнюють нулю). Таким чином, для інструкцій опрацювання даних, на основі відповідної документації [5] було створено відповідну кількість «базових» інструкцій, які працювали зі цифровими константами 0, які записувалися в інструкціях у мінімальні номери регістрів (R0)

або ж виражалися безпосередньо числом.

Використовуючи цей принцип для власних обчислень, було здійснено спробу оцінки енергоспоживання інструкцій, з врахуванням лише коду операції та режиму адресації. Для цього, всім інструкціям було присвоєно порядкові номери (від 0 до 15), які відповідали коду інструкції. Аналогічно було присвоєно порядкові номери (від 0 до 10) режимам адресації, в тому порядку, в якому вони вказані у [4]. Оцінка енергоспоживання проводилася у середовищі MatLab, з використанням пакету nntoolbox, за допомогою якого було створено нейронну мережу. Результати дослідження виявилися як нестабільними (максимальна похибка оцінки змінювалась при кожному запуску), так і неприйнятними (похибка коливалась в діапазоні від 10 % до 1000 %). Отримані результати вимагали проведення додаткового дослідження.

Проведений аналіз отриманих результатів показав, що при оцінці вищезгаданим методом не враховується ряд внутрішніх особливостей роботи інструкції. Їх можна врахувати при аналізі двійкового коду інструкції, який подається на виконання процесорному ядру. Приймаючи до уваги той факт, що дані дослідження проводяться для 32-х бітного режиму роботи мікропроцесора (режим ARM), і всі інструкції мають довжину 32 біти, було проведено аналіз внутрішньої будови інструкцій, та, за допомогою дизасемблера, який входить до комплекту ARM Developer Suite, виділено спільні біти.

Спроба повторення досліду з урахуванням особливостей роботи кожної інструкції дала кращі результати (похибка оцінки енергоспоживання залишалася хоч і високою, проте сталою). На основі аналізу результатів оцінки енергоспоживання, інструкції було розділено на дві групи:

1. В першу групу було відібрано інструкції в тих режимах адресації, в яких дані інструкції мали відносно високе, порівняно з іншими, енергоспоживання. Подальший аналіз показав, що в поданих режимах адресації відбувається звертання до додаткового регістру даних.

2. До другої групи потрапили решта інструкцій.

Результати даного дослідження подано у [6]. Аналіз результатів показав, що за рахунок усунення надлишкової інформації, можна покращити точність оцінки. При цьому надлишкові дані слід досліджувати окремо.

Розширений аналіз даних подано у [3]. Там же подано обґрунтування поділу інструкцій на групи за максимальною кількістю бітів, які беруть участь у формуванні інструкцій. Основну увагу зосереджено на формуванні вибірок для тренування нейронної мережі. Формування відбувається методом пошуку вільного члена лінійної регресії. Незважаючи на те, що проведений аналіз був спрощеним (значення бітів переведено в десяткову форму), даний метод показав найбільш високу точність, порівняно з іншими та дозволив створити методику сортування інструкцій з подібними характеристиками енергоспоживання.

Подальші розрахунки будуть виконуватися без спрощень, що дозволить глибше проаналізувати інструкції за спільними рисами та доцільністю внесення до тієї чи іншої групи.

### 3. Методи відбору інструкцій для аналізу

Першою задачею, яку необхідно вирішити є коректність набору бітів, які будуть використовуватися для подальшого аналізу. Провівши огляд зібраних даних у двійковій формі [7], можна сформулювати критерій для формування матриць регресорів як набору бітів, значення яких змінюються в межах інструкції. Таким чином, матриця, сформована з регресорів, згідно першого методу дозволить оцінити вплив відомих бітів на виконання інструкції. В цьому випадку буде враховано вплив інструкцій відносно базового енергоспоживання інструкції.

Приймаючи до уваги сказане вище, слід провести аналіз та порівняти отримані результати для другої та третьої груп інструкцій опрацювання даних [3]. Варто зазначити, що даний аналіз не має на меті встановити відповідні кореляційні коефіцієнти, а радше оцінити, які інструкції можуть бути об'єднані за енергоспоживанням. Надалі ці групи будуть формувати блоки модульної нейронної мережі, яка буде прогнозувати енергоспоживання кожної інструкції.

Обчислення коефіцієнтів регресії буде проводитися за відомим однокроковим методом

найменших квадратів:

$$\hat{\beta} = (X'X)^{-1}X'y, \quad (1)$$

де  $\hat{\beta}$  — вектор коефіцієнтів регресії;  $X$  — матриця значень відповідних бітів, перший стовпець якої заповнено одиницями, для пошуку вільного члена;  $y$  — вектор значень енергоспоживання.

У таблиці 1 подано результати попередньої обробки даних, а саме виділення змінних бітів в межах однієї інструкції, для третьої групи інструкцій опрацювання даних, на прикладі інструкцій AND та EOR. Також там подано мнемоніку інструкції, її шістнадцяткове представлення та відповідне енергоспоживання. У таблиці 2 подано аналогічну інформацію для другої групи інструкцій опрацювання даних, на прикладі тих самих інструкцій.

Таблиця 1

**Результати попередньої обробки даних третьої групи інструкцій опрацювання даних на прикладі інструкцій AND та EOR**

Мнемоніка	Біти		Шістнадцяткова форма	Енергоспоживання (Дж)
	6	5		
AND R0, R1, R2 LSL R3	0	0	E0010312	2,107524E-09
AND R0, R1, R2 LSR R3	0	1	E0010332	2,092719E-09
AND R0, R1, R2 ASR R3	1	0	E0010352	2,101371E-09
AND R0, R1, R2 ROR R3	1	1	E0010372	2,089939E-09
EOR R0, R1, R2 LSL R3	0	0	E0210312	2,02205E-09
EOR R0, R1, R2 LSR R3	0	1	E0210332	2,048761E-09
EOR R0, R1, R2 ASR R3	1	0	E0210352	2,159229E-09
EOR R0, R1, R2 ROR R3	1	1	E0210372	2,022092E-09

Таблиця 2

**Результати попередньої обробки даних другої групи інструкцій опрацювання даних на прикладі інструкцій AND та EOR**

Мнемоніка	Біти			Шістнадцяткова форма	Енергоспоживання (Дж)
	7	6	5		
AND R0, R1, R2	0	0	0	E0010002	8,55832E-10
AND R0, R1, R2, LSL #1	1	0	0	E0010082	9,1247E-10
AND R0, R1, R2, LSR #1	1	0	1	E00100A2	9,26043E-10
AND R0, R1, R2, ASR #1	1	1	0	E00100C2	8,70551E-10
AND R0, R1, R2, ROR #1	1	1	1	E00100E2	9,06104E-10
AND R0, R1, R2, RRX	0	1	1	E0010062	9,07906E-10
EOR R0, R1, R2	0	0	0	E0210002	9,61443E-10
EOR R0, R1, R2, LSL #1	1	0	0	E0210082	1,051907E-09
EOR R0, R1, R2, LSR #1	1	0	1	E02100A2	1,058837E-09
EOR R0, R1, R2, ASR #1	1	1	0	E02100C2	1,011611E-09
EOR R0, R1, R2, ROR #1	1	1	1	E02100E2	1,070758E-09
EOR R0, R1, R2, RRX	0	1	1	E0210062	1,023248E-09

Слід звернути уваги, що окрім зазначених бітів, які відрізняються в межах однієї інструкції, при подальшому аналізі будуть враховуватися і біти, які відповідають за ідентифікацію інструкції.

#### 4. Результати використання методів відбору інструкцій

В результаті використання методів, описаних у розділі 3, було сформовано матриці регресорів для кожної інструкції. В даних матрицях, в ролі допоміжного регресора, згідно рекомендацій [8] введено одиничний вектор. Таким чином було отримано матриці, коефіцієнти яких подано у таблиці 3. У таблиці 4 подано коефіцієнти регресії, розраховані згідно (1).

Таблиця 3

**Зразки матриць X (матриця регресорів) та вектора Y (регресанда) для інструкцій опрацювання даних, на прикладі інструкцій AND та EOR**

Мнемоніка	$X_1$	$X_2$	$X_3$	$X_4$	Y
AND R0, R1, R2	1	0	0	0	8,55832E-10
AND R0, R1, R2, LSL #1	1	1	0	0	9,1247E-10
AND R0, R1, R2, LSR #1	1	1	0	1	9,26043E-10
AND R0, R1, R2, ASR #1	1	1	1	0	8,70551E-10
AND R0, R1, R2, ROR #1	1	1	1	1	9,06104E-10
AND R0, R1, R2, RRX	1	0	1	1	9,07906E-10
EOR R0, R1, R2	1	0	0	0	9,61443E-10
EOR R0, R1, R2, LSL #1	1	1	0	0	1,051907E-09
EOR R0, R1, R2, LSR #1	1	1	0	1	1,058837E-09
EOR R0, R1, R2, ASR #1	1	1	1	0	1,011611E-09
EOR R0, R1, R2, ROR #1	1	1	1	1	1,070758E-09
EOR R0, R1, R2, RRX	1	0	1	1	1,023248E-09

Таблиця 4

**Коефіцієнти регресії для інструкцій опрацювання даних**

	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$
BIC	6,436E-10	4,9484E-11	1,8791E-11	1,1754E-11
SUB	7,3352E-10	4,276E-11	4,2968E-11	1,5627E-11
MVN	7,6098E-10	3,8655E-11	3,5843E-11	1,6051E-11
CMP	7,7004E-10	3,6348E-11	1,4557E-11	2,5334E-11
SBC	8,0715E-10	3,8883E-11	2,4539E-11	1,5458E-11
TST	8,5387E-10	3,4357E-11	3,0752E-11	2,2617E-11
AND	8,7044E-10	2,1923E-11	-1,6319E-11	3,9173E-11
ADC	8,8572E-10	2,6235E-11	3,749E-11	2,0137E-11
ORR	9,1151E-10	4,2744E-11	3,0128E-11	1,8086E-11
MOV	9,4124E-10	5,0837E-11	2,8363E-11	8,9932E-12
CMN	9,6589E-10	4,4518E-11	1,9925E-11	2,5533E-11
EOR	9,7218E-10	5,5933E-11	-3,449E-12	4,3777E-11
ADD	9,7843E-10	7,5932E-11	8,6628E-12	-1,8273E-12
TEQ	9,8678E-10	4,7359E-11	2,5671E-11	2,1561E-11
RSB	1,0764E-09	3,9585E-12	1,3511E-11	3,7234E-11
RSC	1,1009E-09	3,6833E-11	1,7967E-11	2,1211E-11

#### 5. Архітектура нейронної мережі

Як було сказано у [3] поділ інструкцій на групи може привести до покращення резуль-

татів прогнозу. Тому для верифікації результатів було обрано першу групу, яка містить інструкції BIC, SUB, CMP, MVN, оскільки у попередньому досліді дана група показала дуже хороші результати прогнозування [3]: середня похибка прогнозу становила 1,7 %, а максимальна похибка прогнозу 2,6 %. Надалі для інших інструкцій буде створено відповідні нейронні мережі, які будуть об'єднані у модульну нейронну мережу [9, 10, 11].

З огляду на сказане вище, отримуємо наступну архітектуру нейронної мережі. Вхідний рівень містить 6 нейронів, схований рівень містить 4 нейрони, а вихідний рівень містить 1 нейрон. У ролі нелінійної функції активації обрано багаторівневий персептрон, який є простою та універсальною моделлю оцінки [12, 13]. Вихідне значення трирівневого персептрону можна сформулювати таким чином:

$$y = F_3 \left( \sum_{i=1}^N w_{i3} h_i - T \right),$$

де  $N$  — кількість нейронів схованого рівня,  $w_{i3}$  — ваговий коефіцієнт синапсу від  $i$ -го нейрона прихованого рівня до нейрона вихідного рівня,  $h_i$  — вихідне значення  $i$ -го нейрона,  $T$  — поріг вихідного нейрона, а  $F_3$  — функція активації вихідного нейрона.

Вихідне значення  $j$ -го нейрона, який знаходиться у схованому рівні, знаходимо таким чином:

$$h_j = F_2 \left( \sum_{i=1}^M w_{ij} x_i - T_j \right),$$

де  $w_{ij}$  — ваговий коефіцієнт зв'язку вхідного нейрона та  $j$ -го нейрона схованого рівня,  $x_i$  — вхідні значення, а  $T_j$  — поріг  $j$ -го нейрона. Для прихованого рівня функцією активації є сигмоїда, для вихідного рівня — лінійна функція з коефіцієнтом  $k$  [14].

У ролі алгоритму навчання використовується алгоритм зворотного поширення похибки [15]. Даний алгоритм базується на методі градієнтного зменшення і забезпечує ітераційну процедуру поновлення вагових коефіцієнтів та порогів для кожного вектора  $p$  тренувального набору:

$$\Delta w_{i,j} = -\alpha \frac{\partial E^p(t)}{\partial w_{i,j}(t)};$$

$$\Delta T_j(t) = -\alpha \frac{\partial E^p(t)}{\partial T_j(t)},$$

де  $\alpha$  — коефіцієнт навчання,  $\frac{\partial E^p(t)}{\partial w_{i,j}(t)}$  та  $\frac{\partial E^p(t)}{\partial T_j(t)}$  — градієнти похибки функції  $t$ -ї ітерації тренувального вектора  $p$ ,  $p \in \{1, \dots, P\}$ , де  $P$  — розмір тренувального набору.

Середньоквадратична похибка  $t$ -ї ітерації обчислюється таким чином:

$$E^p(t) = \frac{1}{2} \left( y^p(t) - d^p(t) \right)^2,$$

де для тренувального вектора  $p$ ,  $y^p(t)$  є вихідним значенням  $t$ -ї ітерації, а  $d^p(t)$  — цільове вихідне значення.

В процесі тренування, загальна похибка обчислюється як

$$E(t) = \sum_{p=1}^P E^p(t).$$

Метод найшвидшого сходження для обчислення рівня похибки [Джерело ICNNAI 15] використовується для усунення класичних недоліків алгоритму зворотного поширення похибки. Таким чином, адаптивні коефіцієнти навчання для логарифмічної та лінійної функцій активації відповідно дорівнюють

$$\alpha(t) = \frac{4}{\left(1 + \left(x_i^p(t)\right)^2\right)} \frac{\sum_{j=1}^N \left(\gamma_j^p(t)\right)^2 h_j^p(t) (1 - h_j^p(t))}{\left(\sum_{i=1}^N \left(\left(\gamma_j^p(t)\right)^2 h_j^p(t)\right)\right)^2 (1 - h_j^p(t))^2};$$

$$\alpha(t) = \frac{1}{\sum_{i=1}^N \left(h_i^p(t)\right)^2 + 1},$$

де для  $t$ -ї ітерації  $t$ -го вектора тренувального набору,  $\gamma_j^p(t)$  — похибка  $j$ -го нейрона, а  $h_j^p$  — вхідний сигнал лінійного нейрона.

Похибка  $i$ -го нейрона логарифмічної функції активації визначається згідно наступної формули:

$$\gamma_j^p = \sum_{i=1}^N \gamma_3^p(t) w_{i3}(t) h_i^p(t) (1 - h_j^p(t)),$$

де  $\gamma_3^p(t) = y^p(t) - d^p(t)$  є похибкою вихідного нейрона,  $w_{i3}$  — ваговий коефіцієнт синапсу між нейронами схованого рівня та вихідним нейроном.

## 6. Верифікація результатів відбору за допомогою нейронних мереж

Згідно отриманих регресійних коефіцієнтів, дані було розділено на три групи, за вільним членом. До першої групи увійшли інструкції SUB, BIC, CMP, MVN, вільний член яких знаходиться в діапазоні 6,436E-10...7,7004E-10. Результати оцінки енергоспоживання для даної (другої) групи та порівняння їх з отриманими результатами для цих же інструкцій, у складі третьої групи подано у таблиці 5.

Таблиця 5

Результати оцінки енергоспоживання інструкцій SUB, BIC, CMP, MVN другої і третьої групи

Група інструкцій	Друга	Третя
Архітектура нейронної мережі	6-8-1	8-4-1
Кількість векторів навчального набору	12	8
Кількість векторів оціночного набору	12	8
Середньоквадратична похибка	10 <sup>-8</sup>	1,6 · 10 <sup>-5</sup>
Кількість епох тренування	20785	7134
Середня похибка тренування	0 %	0,6 %
Максимальна похибка тренування	0,1%	0,9 %
Середня похибка прогнозу	2,7 %	1,7 %
Максимальна похибка прогнозу	5,2 %	2,6 %

Як впливає з отриманих результатів, нейронна мережа навчається дуже добре, однак похибка є значно більшою, ніж у випадку оцінки третьої групи. Це погіршення може бути викликане додатковими режимами адресації, які з'являються у даному наборі: операція обробки даних двох регістрів, та операція розвороту з розширенням (RRX). Хоча дані результати є нижчими, аніж результати оцінки відомих математичних моделей, їх потрібно переглянути та додатково проаналізувати на предмет пошуку закономірностей, врахування яких може зменшити похибку оцінки енергоспоживання.

## 7. Висновки

Розглянуто метод оцінки енергоспоживання інструкцій опрацювання даних за допомогою нейронних мереж. У проведених дослідженнях проаналізовано ті інструкції, які належать до другої групи. Отримані результати свідчать про необхідність вдосконалення методу з метою підбору відповідних даних. Такий підбір повинен зменшити в подальшому кількість експериментальних досліджень необхідних для побудови математичних моделей відповідної точності.

З даних результатів впливає і те, що оцінка інших інструкцій буде гіршою, а тому пропонується метод оцінки з врахуванням лише вільного члену регресії вимагає перегляду. Такі дослідження будуть проводитися в подальшому, і матимуть на меті розробку методики оцінки енергоспоживання, яка буде враховувати особливості виконання як інструкції, так і режиму адресації.

## СПИСОК ЛІТЕРАТУРИ

1. Borovyi A. Analysis of CPU's instructions energy consumption device circuits/ A. Borovyi, V. Konstantakos, V. Kochan [et al.] // The fourth IEEE international workshop on Intelligent Data Acquisition and Advancing Computing Systems (IDAACS'2007): Proceedings. — Dortmund, Germany, September 9–11, 2007. — P. 42–47. — ISBN 978-1-4244-1347-8
2. Боровий А. Использование нейронных сетей для оценки энергии выполнения команд микропроцессором / А. Боровий, В. Константакос, В. Кочан [и др.] // Вестник Брестского Государственного Технического Университета, серия: физика, математика и информатика. — 2007. — № 5. — С. 10–13.
3. Borovyi A. Neural-based Prediction of Power Consumption of Data-processing Instructions / A. Borovyi, V. Kochan, V. Turchenko, [et al.] // The 5th International Conference on Neural Networks and Artificial Intelligence (ICNNAI'08): Proceedings. — Minsk, Belarus, May 27–30, 2008. — P. 337–342.
4. «Instruction-level power measurement methodology»: Deliverable 15 / Aristotle University of Thessaloniki; S. Nikolaidis, N. Kavvadias, T. Laopoulos. — AUTH., 2002. — 42 p. — Bibliogr.: p. 42. — № EASY/WP2/DL/P/D15/A1
5. ARM Architecture Reference Manual [electronic resource] / ARM Limited. — 2000. — Issue E. — 811 p. — Режим доступу до довідника: [www.arm.com/miscPDFs/14128.pdf](http://www.arm.com/miscPDFs/14128.pdf)
6. Borovyi A. Predicting Power Consumption of CPU's Core for Base Cost of Data Processing Instructions / A. Borovyi, V. Kochan // The 2008 International Conference on Embedded Systems and Application (ESA 08): Proceedings. — Las-Vegas, NV, USA, July 14 — 17, 2008. — P. 47–51
7. Borovyi A. Using Neural Network for the Evaluation of Power Consumption of Instructions Execution / A. Borovyi, V. Konstantakos, V. Kochan [et al.] // The Fifth International Instrumentation and Measurement Technology Conference (I2MTC'2008): Proceedings. — Vancouver Island, Victoria, British Columbia, Canada, May 12–15, 2008. — P. 676–681.
8. Наследов А. Д. SPSS 15: профессиональный статистический анализ данных / Наследов А. Д. — СПб: Питер, 2008. — 416 с. — ISBN 978-5-388-00193-1
9. Happel B. L. M. Design and evolution of modular character of each instruction. Usage of modular neural network architectures / B. L. M. Happel, and J. M. J. Murre // Neural Networks. — 1994. — P. 985–1004.
10. Куссуль М. Е. Графи модульных нейронных сетей / М. Е. Куссуль // Математичні машини і системи. — 2005 — № 1. — С. 26–38.
11. Галінська А. О. Модульні нейронні мережі: огляд сучасного стану розробок / А. О. Галінська // Математичні машини і системи. — 2003. — № 3, 4. — С. 87–102.
12. Hornik K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, and H. White // Neural Networks. — 1989. — № 2. — С. 359–366.
13. Хайкин С. Нейронные сети: полный курс / С. Хайкин. — М.: Вильямс, 2006. — 1104 с. — ISBN 5-8459-0890-6, 0—13-273350-1
14. Головкин В. А. Нейронные сети: обучение, модели и применение. — М.: Радиотехника, 2001. — 256 с.
15. Rumelhart D. Learning representation by back-propagation errors / D. Rumelhart, G. Hinton, R. Williams // Nature. — № 323. — 1986. — С. 533–536.

Рекомендована кафедрою автоматики та інформаційно-вимірювальної техніки

Надійшла до редакції 21.10.08  
Рекомендована до друку 20.11.08

**Боровий Андрій Модестович** — аспірант.

Кафедра інформаційно-обчислювальних систем та управління, Тернопільський національний економічний університет