

MODEL PREDIKSI KEBANGKRUTAN BERBASIS NEURAL NETWORK DAN PARTICLE SWARM OPTIMIZATION

Isye Arieshanti¹⁾, Yudhi Purwananto¹⁾

¹⁾ Teknik Informatika, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo Surabaya, Jawa Timur
Email : i.arieshanti@if.its.ac.id

ABSTRAK

Kebangkrutan suatu perusahaan dan bank dapat mempengaruhi sistem perekonomian. Karena itulah, pihak-pihak seperti: kreditor, auditor, pemegang saham dan pihak manajemen perusahaan itu sendiri memiliki kepentingan untuk mengetahui kondisi suatu perusahaan yang berhubungan dengan kebangkrutan. Dalam penelitian ini dikembangkan beberapa model klasifikasi untuk memprediksi kebangkrutan suatu perusahaan. Model dikembangkan berdasarkan metode yang berbasis ANN (*Voted Perceptron*, *Stochastic Gradient Descent* dan *Multilayer Perceptron*) dengan metode PSO. Metode-metode yang berbasis ANN bertugas sebagai klasifier dan PSO bertugas sebagai pemilih fitur dan penentu parameter-parameter (*learning rate* dan *epoch*) optimal model. Dari hasil ujicoba dapat disimpulkan bahwa model yang menggabungkan ANN dengan PSO terbukti memiliki performa yang cukup baik, yaitu sekitar 72-75%. Performa terbaik dicapai oleh model *Stochastic Gradient Descent+PSO*, yaitu sebesar 75% dengan jumlah fitur sebanyak 7 fitur. Dengan adanya model prediksi dengan performa yang baik, diharapkan pihak memiliki gambaran yang lebih baik tentang perusahaan yang sedang ditangani. Gambaran tersebut akan membantu pihak-pihak yang berkepentingan dalam mengambil keputusan.

Kata Kunci: *Voted Perceptron*, *Stochastic Gradient Descent*, *Multilayer Perceptron*, *Particle Swarm Optimization*, Prediksi Kebangkrutan

1. PENDAHULUAN

Kebangkrutan suatu perusahaan dan bank dapat mempengaruhi sistem perekonomian. Karena itulah, banyak pihak yang memiliki kepentingan untuk mengetahui kondisi suatu perusahaan yang berhubungan dengan kebangkrutan. Pihak-pihak tersebut diantara adalah kreditor, auditor, pemegang saham dan pihak manajemen perusahaan sendiri [1]. Akan tetapi, teori tentang kebangkrutan masih belum banyak dibahas, sehingga prediksi kebangkrutan menjadi salah satu alat yang penting untuk membantu pihak manajemen, auditor dan kreditor dalam mengevaluasi potensi kebangkrutan dari suatu perusahaan.

Riset untuk prediksi kebangkrutan ini telah dilakukan secara intensif sejak tahun 1960-an [2]. Prediksi kebangkrutan dapat diformulasikan sebagai sebuah model klasifikasi. Model-model klasifikasi ini melibatkan pendekatan metode statistik dan metode *machine learning*. Model klasifikasi yang menggunakan pendekatan statistik meliputi *Univariate Analysis* [3], *Multivariate Discriminant Analysis* [4], *Logistic Regression* [5] dan *Factor Analysis technique* [6]. Selain itu, model klasifikasi untuk memprediksi kebangkrutan juga banyak

dibangun dengan menggunakan metode *machine learning*. Metode-metode tersebut antara lain: *Artificial Neural Network* [7-9], *Support Vector Machines* [10-12] dan *k-Nearest Neighbour* [13-14].

Salah satu metode yang terkenal handal adalah metode *Artificial Neural Network* (ANN). Akan tetapi, dalam pengembangan model klasifikasi menggunakan metode ANN ini ada hal-hal yang harus ditangani secara hati-hati. Sebagai model parametrik, performa ANN tergantung pada beberapa parameter. Parameter-parameter yang penting dalam ANN adalah *learning rate* dan *epoch*. Pemilihan *learning rate* yang terlalu besar akan menyebabkan solusi yang dipilih berada di luar *solution space* yang optimal. Sebaliknya, jika terlalu kecil, maka akan mengakibatkan waktu latihan yang terlalu lama. Demikian juga dengan pemilihan parameter *epoch*. Jika nilai *epoch* terlalu besar, maka akan dihasilkan model yang *overfitting*. Sebaliknya, jika terlalu kecil maka model yang dihasilkan adalah model yang *underfitting*. Karena itulah, untuk menghasilkan model klasifikasi ANN yang optimal, diperlukan optimisasi dalam pemilihan parameter *learning rate* dan *epoch*. Selain parameter, performa suatu model juga dipengaruhi oleh fitur-fitur yang terlibat. Sehingga, pemilihan fitur untuk model klasifikasi

yang akan memprediksi kebangkrutan ini menjadi hal yang penting.

Dengan mempertimbangkan pentingnya pemilihan parameter dan fitur yang akan digunakan dalam pengembangan model prediksi kebangkrutan, maka pada penelitian ini dikembangkan sebuah model baru yang menggabungkan antara ANN dan *Particle Swarm Optimization* (PSO). Metode ANN digunakan untuk mengklasifikasikan apakah suatu perusahaan berpotensi sebagai perusahaan yang akan bangkrut atau tidak. Metode ANN yang akan dikembangkan terdiri dari 3 varian, antara lain *Voted Perceptron*, *Stochastic Gradient Descent* dan *Multilayer Perceptron*. Sedangkan metode PSO berperan sebagai metode optimasi yang menentukan nilai *learning rate*, *epoch* dan fitur-fitur yang akan digunakan dalam model klasifikasi. Performa dari metode gabungan ini akan dibandingkan dengan performa model klasifikasi yang berbasis ANN saja.

2. DATA DAN METODE

Pada bagian ini akan dijelaskan tentang data dan metode yang digunakan untuk membangun model prediksi kebangkrutan. Pada bagian selanjutnya akan dideskripsikan tentang dataset yang digunakan sebagai data latih dan data testing dalam pengembangan model prediksi kebangkrutan ini. Kemudian akan dijelaskan metode ANN yang terdiri dari *Voted Perceptron*, *Stochastic Gradient Descent*, *Multilayer Perceptron* dan *Particle Swarm Optimization*.

2.1 Dataset

Dataset yang digunakan adalah dataset dari Wieslaw dan rekan [15]. Dataset ini terdiri dari 240 data. Dimana 240 data itu terbagi menjadi 112 data positif dan 128 data negatif. Data positif disini merepresentasikan data dari perusahaan yang bangkrut dan data negatif merepresentasikan data dari perusahaan yang sehat. Masing-masing data memiliki 30 dimensi/fitur. Masing-masing fitur merupakan nilai rasio finansial dari perusahaan tersebut.

2.2 Voted Perceptron

Algoritma ini merupakan sebuah algoritma perceptron untuk melakukan klasifikasi linier dengan memaksimalkan margin antara dua kelas data. Dalam model ini, setiap data direpresentasikan sebagai (x, y) , dimana $x_1, \dots, x_n \in X$ adalah vektor fitur yang merepresentasikan data-data perusahaan dan $y_1, \dots, y_n \in \{+1, -1\}$ adalah label dari masing-masing data (perusahaan yang berpotensi bangkrut atau tidak).

Pada saat proses latih, model dilatih untuk mendefinisikan beberapa perceptron terbobot. Beberapa perceptron terbobot itu didefinisikan berdasarkan beberapa vektor prediktor v_1, \dots, v_k .

Setiap vektor prediktor memiliki dimensi yang sama dengan vektor fitur. Di awal proses latih, dilakukan inialisasi sebuah vektor prediktor $v_1=0$. Selanjutnya vektor tersebut di gunakan dalam sebuah perceptron untuk memprediksi label data perusahaan, dengan persamaan 1 [16].

$$y = \text{sign} (v_1 \cdot x) \quad (1)$$

Jika nilai y sama dengan label data x_1 maka perhitungan perceptron untuk sample berikutnya tetap memanfaatkan vektor prediktor v_1 . Selain itu, bobot $v_1 (c_1)$ ditingkatkan berdasarkan banyaknya jumlah iterasi saat v_1 selalu benar dalam memprediksi label sample berikutnya. Tetapi, jika hasil persamaan (1) berbeda dengan label data x_n , maka didefinisikan vektor prediktor baru dengan persamaan (2) [16].

$$v_{k+1} = v_k + y_n \cdot x_n \quad (2)$$

Dalam proses training ini, algoritma terus diiterasi menggunakan semua sample, sampai didapatkan vektor prediktor yang selalu benar dalam memprediksi label data kebangkrutan perusahaan. Vektor prediktor beserta bobotnya yang dihasilkan dalam proses training ini disimpan untuk digunakan dalam proses prediksi atau klasifikasi.

Dalam proses prediksi, sebuah data perusahaan x akan diberi label positif (berpotensi untuk bangkrut) atau negatif (perusahaan yang sehat) berdasarkan voting perceptron. Proses voting perceptron memanfaatkan vektor prediktor dan bobotnya $(v_1, c_1), \dots, (v_k, c_k)$ yang tersimpan dari proses latih. Dimana c_i adalah jumlah voting yang dimiliki oleh setiap vektor prediktor dan nilai k adalah banyaknya vektor prediktor yang dihasilkan pada saat latih. Perhitungan prediksinya seperti yang didefinisikan oleh persamaan (3) [16].

$$y(x) = \text{sign} \left(\sum_{i=1}^k c_i \cdot \text{sign} (v_i \cdot x) \right) \quad (3)$$

2.3 Stochastic Gradient Descent

Metode Stochastic Gradient Descent adalah suatu metode yang memanfaatkan gradient descent secara incremental untuk mencari bobot vektor yang paling fit dengan data training. Proses update bobot secara incremental dilakukan berdasarkan penghitungan error dari setiap data latih. Arsitektur jaringannya menggunakan 2 layer yaitu layer *input* dan layer *output* [17].

Dalam proses prediksi, sebuah data perusahaan x akan diberi label positif (berpotensi untuk bangkrut) atau negatif (perusahaan yang sehat) berdasarkan persamaan (4). Pada persamaan (4) w_i adalah vektor bobot yang dihasilkan dari proses latih, w_0 adalah *bias* dan n adalah dimensi fitur data. Fungsi aktivasi yang digunakan adalah fungsi sigmoid.

$$y(x) = \text{sign} \left(\frac{1}{1 + \exp \left(- \sum_{i=1}^n w_i \cdot x_i + w_0 \right)} \right) \quad (4)$$

Pada saat proses latih, vektor bobot pada awalnya diinisialisasi dengan nilai antara [0,1] secara random. Selanjutnya vektor bobot diupdate untuk setiap data latih dengan persamaan (5). Pada persamaan (5) η merupakan simbol learning rate, o adalah nilai yang dihitung berdasarkan Neural Network Stochastic Gradient Descent dan t adalah label data latih yang sebenarnya [17]. Proses update vektor bobot dilakukan secara berulang sesuai dengan nilai epoch yang ditentukan.

$$w_i = w_i + \eta(t - o)x_i \quad (5)$$

2.4 Multilayer Perceptron

Metode Multilayer Perceptron adalah metode ANN yang memiliki arsitektur jaringan yang terdiri sekurang-kurang 3 layer. Ketiga layer tersebut antara lain layer *input*, layer *hidden* dan layer *output*. Sama seperti metode-metode ANN yang lain, metode ini bertujuan untuk mendapatkan vektor bobot yang paling *fit* dengan data latih. Dalam melakukan update vektor bobot, metode ini menggunakan algoritma *backpropagation*. Sama seperti metode *Stochastic Gradient Descent*, pada metode ini output dari masing-masing layer menggunakan fungsi aktivasi Sigmoid.

Dalam proses prediksi, sebuah data perusahaan x akan diberi label positif (berpotensi untuk bangkrut) atau negatif (perusahaan yang sehat) berdasarkan persamaan (6). Pada persamaan (6) w_i adalah vektor bobot yang dihasilkan dari proses latih, w_0 adalah *bias* dan n adalah dimensi fitur data.

$$y(x) = \text{sign} \left(\frac{1}{1 + \exp \left(- \sum_{i=1}^n w_i \cdot x_i + w_0 \right)} \right) \quad (6)$$

Pada saat proses latih, bobot vektor diupdate dalam dua tahap. Tahap awal pada proses latih sama dengan yang dilakukan pada algoritma Gradient Descent, yaitu inialisasi nilai awal untuk vektor bobot, baik vektor bobot di *layer input* maupun vektor bobot di *layer hidden*. Selanjutnya dilakukan *forward propagation* untuk menghitung *output* dari *network*. Perhitungan dimulai dari *layer input*, *layer hidden* dan *layer output*. Setelah didapatkan nilai *output* di *layer output* o_k dan nilai *output* di *layer hidden* o_h , selanjutnya dilakukan *backward*

propagation untuk menghitung *error* di *layer output* δ_k (persamaan 7) dan *layer hidden* δ_h (persamaan 8). Pada persamaan (8), w_{kh} adalah nilai bobot yang menghubungkan unit hidden dan unit output yang sesuai [17].

$$\delta_k = o_k (1 - o_k) (t_k - o_k) \quad (7)$$

$$\delta_h = o_h (1 - o_h) \sum_{k \in \text{output}} w_{kh} \cdot \delta_k \quad (8)$$

Berdasarkan *error* yang diperoleh, dilakukan update vektor bobot di *layer input* (persamaan 9) dan vektor bobot di *layer hidden* (persamaan 10). Proses ini dilakukan secara iteratif. Banyaknya iterasi ditentukan berdasarkan jumlah epoch yang ditetapkan [17].

$$w_{ih} = w_{ih} + \eta \delta_h x_i \quad (9)$$

$$w_{kh} = w_{kh} + \eta \delta_k o_h \quad (10)$$

2.5 Particle Swarm Optimization (PSO)

Metode PSO merupakan metode optimasi yang diinspirasi oleh perilaku ikan dan burung yang bergerak secara berkelompok. Dalam PSO, setiap individu diperlakukan sebagai partikel dalam space d -dimensi dan partikel-partikel tersebut memiliki posisi dan kecepatan. Sebuah partikel diimplementasikan sebagai sebuah vektor. Posisi sebuah partikel ke- i direpresentasikan sebagai $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ dan kecepatan partikel ke- i direpresentasikan sebagai $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$. Proses update kecepatan setiap partikel diformulasikan seperti pada persamaan (11) dan update posisi masing-masing partikel diformulasikan seperti pada persamaan (12) [18].

$$v_{i,j}^{n+1} = w \cdot v_{i,j}^n + c_1 \cdot r_1 (p_{i,j}^n - x_{i,j}^n) + c_2 \cdot r_2 (p_{g,j}^n - x_{i,j}^n) \quad (11)$$

$$x_{i,j}^{n+1} = x_{i,j}^n + v_{i,j}^{n+1} \quad j = 1, 2, \dots, d \quad (12)$$

Pada persamaan (11), vektor $p_i^n = (p_{i,1}^n, p_{i,2}^n, \dots, p_{i,d}^n)$ merupakan posisi vektor ke- i terbaik yang didapatkan sampai iterasi ke- n iterasi. Sedangkan $p_g^n = (p_{g,1}^n, p_{g,2}^n, \dots, p_{g,d}^n)$ merupakan posisi terbaik dari semua partikel (dalam populasi). Simbol lain yang digunakan pada persamaan (11) adalah w , c_1 , r_1 , c_2 dan r_2 . Nilai w , r_1 , r_2 , c_1 dan c_2 secara berturut-turut adalah nilai inersia bobot, random dalam rentang [0,1] dan nilai konstanta. Penentuan posisi terbaik setiap partikel dan posisi terbaik partikel dalam populasi ditentukan berdasarkan nilai *fitness function* [18].

Dalam model prediksi yang dikembangkan pada penelitian ini, dimensi partikel adalah 32.

Dimensi ke-1 sampai ke-30 adalah fitur yang akan dioptimasi. Jumlah ini sesuai dengan jumlah fitur dataset kebangkrutan. Sedangkan 2 dimensi terakhir adalah nilai epoch dan nilai learning rate metode ANN (*Voted Perceptron*, *Stochastic Gradient Descent* dan *Multilayer Perceptron*). Nilai *fitness function* merupakan nilai akurasi dari masing-masing metode.

3. SKENARIO UJI COBA

Uji coba model dalam penelitian ini dilakukan dalam tiga skenario. Skenario pertama, model dibangun dengan menggunakan gabungan *Voted Perceptron* sebagai klasifier dan PSO sebagai metode optimasi yang menentukan fitur dan parameter epoch. Parameter epoch dicari dalam rentang [1, 100]. Performa dari model ini dibandingkan dengan performa *Voted Perceptron* yang menggunakan semua fitur (30 fitur).

Skenario kedua dilakukan dengan membangun model gabungan antara *Stochastic Gradient Descent* dengan PSO. Parameter learning rate dicari pada rentang [0.001, 1] dan parameter epoch dicari pada rentang [1, 5000]. Sama seperti pada skenario sebelumnya, performa dari model ini akan dibandingkan dengan model *Stochastic Gradient Descent* yang menggunakan semua fitur.

Skenario terakhir dilakukan dengan membangun model gabungan antara *Multilayer Perceptron* dengan PSO. Parameter learning rate dicari pada rentang [0.001, 1] dan parameter epoch dicari pada rentang [1, 1000]. Sama seperti pada skenario yang lainnya, performa dari model ini akan dibandingkan dengan model *Multilayer Perceptron* yang menggunakan semua fitur.

Performa semua model diukur berdasarkan nilai accuracy dari 10-fold cross-validation. Hal ini dapat direpresentasikan bahwa nilai akurasinya merupakan nilai rata-rata akurasi dari 10 *fold*. Definisi accuracy seperti pada persamaan (13) berikut:

$$accuracy = \frac{TP + TN}{\text{jumlah seluruh data}} \quad (13)$$

Dimana TP dan TN adalah jumlah data positif dan data negatif yang diprediksi secara tepat oleh klasifier. Pengembangan model dilakukan dengan menggunakan bahasa pemrograman Java.

4. HASIL DAN ANALISA UJI COBA

Hasil uji coba dari masing-masing skenario ditampilkan pada tabel 1. Dari tabel terlihat bahwa semua model yang menggunakan gabungan metode ANN yang dioptimasi dengan menggunakan PSO, memiliki nilai akurasi yang lebih tinggi jika dibandingkan dengan nilai akurasi model yang tidak menggunakan seleksi fitur.

Perbedaan akurasi antara model yang melakukan seleksi fitur dengan model yang tanpa

melakukan seleksi fitur sekitar 11-19%. Perbandingan akurasi untuk skenario pertama, akurasi model *Voted Perceptron + PSO* berbeda signifikan secara statistik dengan akurasi model *Voted Perceptron*. Uji statistik menggunakan t-test dengan level signifikansi $\alpha=0.05$. Uji statistik menunjukkan hasil yang sama pula untuk skenario *Stochastic Gradient Descent + PSO* dibandingkan dengan *Stochastic Gradient Descent* dan *Multilayer Perceptron + PSO* dibandingkan dengan *Multilayer Perceptron* saja.

Jika perbandingan akurasi dilakukan untuk semua model, terlihat model *Stochastic Gradient Descent + PSO* mencapai tingkat akurasi yang paling tinggi, sebesar 75%. Sedangkan model dengan nilai akurasi terendah adalah model *Voted Perceptron* sebesar 53%. Nilai akurasi ini mendekati nilai prediksi random. Lebih jauh lagi, jika dibandingkan dengan metode yang berbasis *Stochastic Gradient Descent* dan *Multilayer Perceptron*, metode *Voted Perceptron* selalu mencapai nilai yang lebih rendah. Hal ini mungkin terjadi karena metode *Voted Perceptron* tidak menggunakan *learning rate*. Dimana dengan adanya *learning rate*, model mampu mengeksplorasi solution space secara lebih baik. Akan tetapi, jika data bersifat *linearly separable* ada kemungkinan bagi metode *Voted Perceptron* untuk mencapai performa yang sebanding dengan varian ANN yang lain. Terlebih lagi, salah satu kelebihan dari metode *Voted Perceptron* adalah kesederhanaan algoritma sehingga mudah untuk diimplementasikan.

Jika dilihat dari sisi jumlah fitur, model-model yang menggabungkan varian-varian ANN dengan PSO mampu mencapai akurasi yang lebih tinggi dengan menggunakan jumlah fitur yang jauh lebih sedikit. Jumlah fitur yang digunakan hampir 20% saja dari keseluruhan jumlah fitur. Hubungan antara pengurangan jumlah fitur dengan peningkatan akurasi adalah sebagai berikut. Fitur-fitur yang terpilih oleh metode PSO merupakan fitur penting dan bersifat diskriminatif. Sedangkan fitur yang direduksibisa jadi merupakan fitur yang memiliki nilai sama antara data positif dan data negatif, sehingga cenderung bersifat noise.

Pengurangan jumlah fitur ini tentunya akan menguntungkan dari sisi waktu komputasi. Karena model yang menggunakan fitur lebih sedikit secara otomatis akan menggunakan waktu komputasi yang lebih cepat. Memang dibutuhkan waktu lebih untuk menemukan fitur-fitur dan parameter-parameter yang lebih optimal, akan tetapi jika model sudah terbangun, proses untuk memprediksi apakah suatu perusahaan berpotensi bangkrut atau tidak dapat dilakukan secara lebih cepat.

Tabel 1. Perbandingan Akurasi dari beberapa model yang dikembangkan

Model	Rata-rata Akurasi	Jumlah Fitur
Voted Perceptron + PSO; epoch=3	72.1%	6
Voted Perceptron epoch=3	53%	30
Stochastic Gradient Descent + PSO; $\eta=1$, epoch=199	75%	7
Stochastic GradientDescent ; $\eta=1$, epoch=199	60%	30
Multilayer Perceptron + PSO; $\eta=0.4$, epoch=304	73%	7
Multilayer Perceptron $\eta=0.4$, epoch=304	62%	30

5. KESIMPULAN

Dalam penelitian ini dikembangkan beberapa model klasifikasi untuk memprediksi kebangkrutan suatu perusahaan. Dengan adanya model prediksi dengan performa yang baik, diharapkan pihak kreditor, auditor, pemegang saham dan pihak manajemen perusahaan itu sendiri memiliki gambaran yang lebih baik tentang perusahaan yang sedang ditangani. Gambaran tersebut akan membantu pihak-pihak yang berkepentingan dalam mengambil keputusan. Untuk membangun model prediksi dengan performa yang baik itulah, maka pada penelitian ini dirancang beberapa model klasifikasi yang menggabungkan metode berbasis ANN (*Voted Perceptron*, *Stochastic Gradient Descent* dan *Multilayer Perceptron*) dengan metode PSO. Metode-metode yang berbasis ANN bertugas sebagai klasifier dan PSO bertugas sebagai pemilih fitur dan penentu parameter-parameter (*learning rate* dan *epoch*) optimal model.

Dari hasil ujicoba dapat dilihat bahwa model yang menggabungkan ANN dengan PSO terbukti memiliki performa yang cukup baik, yaitu sekitar 72-75%. Performa terbaik dicapai oleh model *Stochastic Gradient Descent+PSO*, yaitu sebesar 75% dengan jumlah fitur sebanyak 7 fitur. Sehingga sapat disimpulkan bahwa untuk membangun sebuah model klasifikasi yang mampu memprediksi apakah sebuah perusahaan kemungkinan akan bangkrut atau tidak, diperlukan gabungan metode klasifikasi yang handal dan juga metode optimasi yang bertugas untuk memilih fitur dan menentukan nilai parameter model. Kesimpulan yang lain adalah fitur-fitur yang dipilih oleh PSO dapat meningkatkan akurasi karena fitur tersebut merupakan fitur yang bersifat diskriminatif.

6. DAFTAR PUSTAKA

- [1] R.L. Wilson, R. Sharda. 1994. "Bankruptcy prediction using neural networks". **Decision Support Systems** 11 545–557.
- [2] E.I. Altman. 1968. "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy". **Journal of Finance** 23: 89–609.
- [3] W.H. Beaver. 1966. "Financial ratios as predictors of failure". **Journal of Accounting Research** 4:71–111.
- [4] E.I. Altaian. 1968. "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy". **Journal of Finance** 23, 4: 589–609.
- [5] J.A. Ohlson. 1980. "Financial ratios and the probabilistic prediction of bankruptcy". **Journal of Accounting Research** 109–131.
- [6] R.C. West. 1985. "A factor-analytic approach to bank condition". **Journal of Banking & Finance** 9 (2): 253–266.
- [7] C.F. Tsai. 2008. "Financial decision support using neural networks and support vector machines". **Expert Systems** 25 (4): 380–393.
- [8] G. Zhang. 1999. "Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis". **European Journal of Operational Research** 116 (1): 16–32.
- [9] P. Ravisankar, V. Ravi. 2010. "Financial distress prediction in banks using Group Method of Data Handling neural network, counter propagation neural network and fuzzy ARTMAP". **Knowledge-Based Systems** 23 (8) : 823–831.
- [10] K.S. Shin, T.S. Lee, H.J. Kim. 2005. "An application of support vector machines in bankruptcy prediction model". **Expert Systems with Applications** 28 (1): 127–135.
- [11] J.H. Min, Y.C. Lee. 2005. "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters". **Expert Systems with Applications** 28 (4): 603–614.
- [12] L. Fengyi, Y. Ching-Chiang, L. Meng-Yuan. 2011 "The use of hybrid manifold learning and support vector machines in the prediction of business failure". **Knowledge-Based Systems** 24 (1): 95–101.
- [13] A.Y.N. Yip. 2004. "Predicting business failure with a case-based reasoning approach". **Proceedings of 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems: KES**, Wellington, New Zealand. September 3215/2004, Part III, , pp. 20–25.
- [14] C.-S. Park, I. Han. 2002. "A case-based reasoning with the feature weights derived by

- analytic hierarchy process for bankruptcy prediction”. **Expert Systems with Applications** 23 (3): 255–264.
- [15] P. Wieslaw, 2004. “Application of discrete predicting structures in an early warning expert system for financial distress”. **Ph.D. Thesis**, Szczecin Technical University, Szczecin.
- [16] Y. Freund and R. E. Schapire. 1999. “Large margin classification using the perceptron algorithm”. **Machine Learning**, 37:277-296
- [17] Tom M.Mitchell. 1997. “Artificial Neural Network” in **Machine Learning**, McGraw-Hill, International Edition, Singapore.
- [18] R.C. Eberhart, J. Kennedy. 1995. “A new optimizer using particle swarm theory” **Proceeding of Sixth International Symposium on Micro Machine and Human Science**, Nagoya, pp. 39–43.