

ỨNG DỤNG MẠNG NƠON TÍNH TOÁN VÀ DỰ ĐOÁN CHẤT LƯỢNG NƯỚC ĐẦU RA CỦA HỆ THỐNG XỬ LÝ NƯỚC THẢI CỦA NHÀ MÁY SỮA CÔ GÁI HÀ LAN

TS. Nguyễn Kỳ Phùng, CN Đậu Thị Dung
Khoa Môi trường - Trường Đại học Khoa học Tự nhiên
Đại học Quốc gia TP.HCM

Mạng nơon mô hình cấu trúc được sử dụng khai thác rộng rãi để nghiên cứu, tính toán trong nhiều lĩnh vực khác nhau. Trong lĩnh vực môi trường, lý thuyết mạng nơon được dùng để tính toán và dự đoán kết quả đầu ra của hệ thống xử lý nước thải hoặc đánh giá khả năng tái sử dụng nguồn nước sau khi xử lý... Trong bài báo này tác giả muốn giới thiệu với bạn đọc mô hình mạng nơon được phát triển để tính toán và dự đoán chất lượng nước đầu ra của hệ thống xử lý nước thải Nhà máy sữa Cô gái Hà Lan. Các thông số được quan trắc hằng ngày bao gồm: độ kiềm pH, lưu lượng Q , nhu cầu oxy hóa học COD và nồng độ chất rắn lơ lửng SS. Cấu trúc của mô hình mạng nơon được xác định thông qua các bước luyện và kiểm tra mô hình. Thực tế cho thấy sai số trung bình của hai chương trình không chênh lệch nhiều. Như vậy kết quả tính toán các thông số chất lượng nước đầu ra của mô hình mạng nơon nhân tạo là chấp nhận được.

1. Giới thiệu

Nước thải chứa rất nhiều tạp chất có bản chất khác nhau, mục đích của việc xử lý nước thải là khử các tạp chất đó sao cho nước sau khi xử lý đạt tiêu chuẩn chất lượng. Các tiêu chuẩn đó thường phụ thuộc vào mục đích và cách thức sử dụng: nước sẽ được tái sử dụng hay thải thẳng vào các nguồn tiếp nhận nước và với những công nghệ xử lý khác nhau, chất lượng nước đầu ra sẽ đạt tiêu chuẩn nào? Liệu ta có thể biết được các thông số chất lượng nước sau khi xử lý mà không cần tiến hành thí nghiệm? giải pháp ứng dụng lý thuyết mạng nơon là một trong những phương pháp có thể giải quyết được vấn đề trên. Việc tính toán kết quả chất

lượng nước đầu ra của hệ thống xử lý nước thải có thể góp phần giúp các nhà máy xử lý nước thải quản lý chất lượng nước sau khi xử lý đồng thời có thể đáp ứng và phân phối nước cho các nhu cầu sử dụng khác nhau. Vì vậy, bài báo này sẽ sử dụng cập đến vấn đề ứng dụng cấu trúc mạng trí tuệ nhân tạo (mạng nơon) tính toán kết quả đầu ra của hệ thống xử lý nước thải.

2. Nội dung

Trong nhiều thập niên qua, mạng nơon nhân tạo - một mô hình tính toán mô phỏng bộ não con người - đã được sử dụng có hiệu quả trong các lĩnh vực như: trí tuệ nhân tạo, nhận dạng chữ viết, xử lý ảnh, kinh tế, y học, hóa học, điều khiển, thủy văn... và những thành quả mang lại là

rất to lớn. Tuy nhiên, trong lĩnh vực môi trường tại Việt Nam hiện nay vẫn còn ít nghiên cứu cụ thể về vấn đề này.

Chính vì vậy bài báo tập trung nghiên cứu hai nội dung chính:

- Tìm hiểu mạng nơon nhân tạo và những ứng dụng của nó.

- Xây dựng chương trình tính toán kết quả đầu ra của hệ thống xử lý nước thải.

Mạng nơon nhân tạo (Artificial Neural Network - ANN) bao gồm các "nút" (đơn vị xử lý, nơon) được nối với nhau bởi các liên kết nơon. Mỗi liên kết kèm theo một trọng số nào đó, có thể xem các trọng số là phương tiện để lưu thông tin dài hạn trong mạng nơon và nhiệm

vụ của quá trình luyện mạng (hay quá trình học) là cập nhật lại các trọng số khi có thêm các thông tin về các mẫu học. Hay nói một cách khác, các trọng số được điều chỉnh sao cho kết quả mà mạng nơron tính được phù hợp với kết quả quan trắc. Một mạng nơron cơ bản thường có: một lớp nhập, một lớp ẩn và một lớp xuất. Lớp nhập nhận giá trị đầu vào (ở đây là các thông số của nước thải trước khi xử lý). Lớp xuất tính toán kết quả đầu ra (các thông số của nước thải sau khi xử lý).

3. Các bước xây dựng mô hình ANN

* Chia dữ liệu

Dữ liệu sẵn có được chia thành hai tập hợp: tập mẫu luyện và tập mẫu kiểm tra. Mạng dùng tập mẫu luyện để học hỏi các mối quan hệ giữa biến đầu vào và các biến đầu ra, tập kiểm tra để kiểm tra mô hình. Thông thường số lượng mẫu luyện chiếm 70% toàn bộ tập mẫu, nếu số lượng mẫu luyện ít, mạng sẽ không có đủ dữ kiện để tổng quát hóa cho toàn bộ tập mẫu.

* Tiền xử lý dữ liệu

Trong quá trình tính toán của mạng nơron ta có sử dụng một số hàm toán học như hàm truyền logistic:

$$y = \frac{1}{1 + e^{-x}}$$

Hàm này có cận trên là 1 cận dưới là 0, do đó các giá

trị đầu vào phải được chuẩn hóa về khoảng giới hạn của hàm truyền (0,1). Tuy nhiên, 0 và 1 chỉ là những giới hạn mà hàm truyền tiến gần đến được chứ không bao giờ thực sự đạt được. Do đó, trong thực tế người ta thường chuẩn hóa các giá trị đầu vào nằm trong đoạn [0.1, 0.9].

Công thức chuẩn hóa thường được sử dụng nhất là chuẩn hóa Min - Max.

$$y' = \frac{y - \text{Min1}}{\text{Max1} - \text{Min1}} * (\text{Max2} - \text{Min2}) + \text{Min2} \quad (1)$$

Trong đó:

y: giá trị cũ.

y': giá trị mới.

Min1, Max1: giá trị cực tiểu và cực đại của miền giá trị cũ.

Min2, Max2: giá trị cực tiểu và cực đại của miền giá trị mới.

* Xác định các biến

Đây là một nhiệm vụ quan trọng bởi vì khi ta xác định được các biến nhập hợp lý, mạng nơron sẽ dễ dàng thành công trong việc cho kết xuất thực gần với kết quả mong muốn nhất.

* Xác định cấu trúc mạng

- Sử dụng loại mạng nào, mạng lan truyền tiến hay mạng hồi quy?

- Xác định cấu trúc hình học của mạng. Mạng có kích thước lớn hay nhỏ? Mạng có bao nhiêu lớp ẩn? Số nút trên lớp ẩn là bao nhiêu?

Quá trình học (luyện mạng)

- Xác định các luật học:

hàm truyền, hàm lỗi, hệ số học...

- Khởi tạo trọng số ngẫu nhiên.

- Cho các tín hiệu truyền đi trong mạng.

- Tính sai số và lan truyền ngược sai số.

- Tìm bộ trọng số tối ưu.

* Kiểm tra - chuẩn hóa mô hình

Cho mô hình tính toán với các mẫu mà nó chưa từng học

trước đó, nếu kết quả cho sai số lớn thì tiến hành xây dựng lại mô hình mạng.

* Ứng dụng:

Sau đây là ví dụ minh họa cho việc ứng dụng mạng nơron để tính toán kết quả đầu ra của hệ thống xử lý nước thải. Dữ liệu gồm 96 mẫu lấy từ hệ thống xử lý nước thải của nhà máy sữa Cô gái Hà Lan, Thủ Đức, Tp.HCM. Bốn biến đầu vào được sử dụng bao gồm: pH, Q, COD và SS; một biến đầu ra là COD tương ứng với 1 nút xuất.

Bước 1: Tiền xử lý dữ liệu

* Các biến nhập

$$P1 \frac{pH - \min(pH)}{\max(pH) - \min(pH)} * 0.8 + 0.1$$

$$P2 \frac{Q - \min(Q)}{\max(Q) - \min(Q)} * 0.8 + 0.1$$

$$P3 \frac{COD_{120} - \min(COD_{120})}{\max(COD_{120}) - \min(COD_{120})} * 0.8 + 0.1$$

$$P4 \frac{SS - \min(SS)}{\max(SS) - \min(SS)} * 0.8 + 0.1$$

* **Biến xuất:**

$$\frac{COD_{ra} - \min(COD_{vao}, COD_{ra})}{\max(COD_{vao}, COD_{ra}) - \min(COD_{vao}, COD_{ra})} * 0.8 + 0.1$$

Gán sai số trên toàn bộ tập mẫu luyện E = 0. Đối với mỗi mẫu thứ m (m=1,2,...,70) trong tập mẫu luyện làm từ bước 2 đến bước 5.

Lan truyền tiến

* **Bước 2:**

Mỗi nút nhập X_i (i=1,2,3,4) nhận một tín hiệu đầu vào P_i và gửi các tín hiệu này đến tất cả các nút ẩn trong lớp ẩn.

Với mỗi nút ẩn H_j tổng trọng số của nó sẽ được tính.

$$TongTrongSo_An(j) = \sum_{i=1}^4 w_{ij} * P_i \quad (j=1,2,3) \quad (2)$$

* **Bước 3:**

Tính đầu ra của mỗi nút trên lớp ẩn (kết xuất nút ẩn) thông qua hàm truyền logistic:

$$H(j) = KetXuat_An(j) = f(TongTrongSo_An(j)) = \frac{1}{1 + e^{-(TongTrongSo_An(j))}} \quad (j=1,2,3) \quad (3)$$

* **Bước 4:** Sau khi tính đầu ra của tất cả các nút ẩn, kết quả này sẽ được gửi đến tất cả các nút xuất trên lớp xuất. Tổng trọng số của tất cả các nút trên lớp xuất sẽ được tính:

$$TongTrongSo_Xuat(k) = \sum_{j=1}^3 W_{jk} * H(j) \quad (k=1; j=1,2,3) \quad (4)$$

Tính đầu ra của mỗi nút trên lớp xuất (kết xuất nút xuất) thông qua hàm truyền logistic:

$$O(k) = KetXuat_Xuat(k) = f(TongTrongSo_Xuat(k)) = \frac{1}{1 + e^{-(TongTrongSo_Xuat(k))}} \quad (k=1)$$

* **Bước 5:**

Tính sai số giữa đầu ra mong muốn và đầu ra thực tế của nút xuất thứ k, của mẫu thứ m. Ở đây k=1 nên:

$$E_m = (Y_m(k) - O(k))^2 \quad (6)$$

$Y_m(k)$: đầu ra mong muốn của nút xuất thứ k đối với mẫu luyện thứ m. Nếu đã duyệt xong toàn bộ tập mẫu luyện chuyển qua bước 6

* **Bước 6:**

Tính tổng sai số trung bình bình phương trên toàn bộ tập mẫu luyện $E = \frac{1}{2} \sum_{m=1}^{73} E_m$ (m=73) (7)

So sánh sai số E với E_{max} : Nếu $E > E_{max}$ thì qua bước 7. Nếu $E \leq E_{max}$ thì ngừng luyện.

Lan truyền ngược sai số

* **Bước 7 :** Cập nhật lại trọng số của các nút trên lớp xuất.

Tính Delta xuất của nút xuất thứ k đối với mỗi mẫu luyện thứ m:

$$\delta_{km} = O(k) * (1 - O(k)) * (Y_m(k) - O(k)) \quad (8)$$

Cập nhật lại trọng số liên kết giữa nút ẩn thứ j và nút xuất thứ k: $\Delta W_{jk} = \eta \sum_{m=1}^{73} \delta_{km} H_{jm}$ (9)

$W_{jk}(\text{mới}) = W_{jk}(\text{cũ}) + \Delta W_{jk}$ Trong đó: $\eta = 0,02$ là hệ số học;

$\alpha = 0,6$ là hệ số moment

* **Bước 8:**

Cập nhật lại trọng số của các nút trên lớp ẩn. Tính delta ẩn của nút ẩn thứ j đối với mẫu luyện thứ m: $\delta_{jm} = H_{jm}(1 - H_{jm}) \sum_{k=1}^{\text{outs}} W_{jk} \delta_{km}$ (10)

H_{jm} : đầu ra của nút ẩn thứ j đối với mẫu thứ m .

δ_{km} : delta xuất của nút xuất thứ k .

W_{jk} : trọng số liên kết giữa các nút trên lớp ẩn và trên lớp xuất.

O_{outs} : tổng các nút xuất. Trong ví dụ này chỉ sử dụng 1 nút xuất nên delta của nút ẩn thứ j sẽ được tính như sau: $\delta_{jm} = H_{jm}(1-H_{jm}) \sum_{k=1}^{outs} W_{jk} \delta_{km}$ (11)

Cập nhật trọng số liên kết giữa nút nhập thứ i và nút ẩn thứ j : $\delta_{km} = \Delta w_{ij} = \eta \sum_{m=1}^{73} \delta_{jm} P_{im}$ (12)

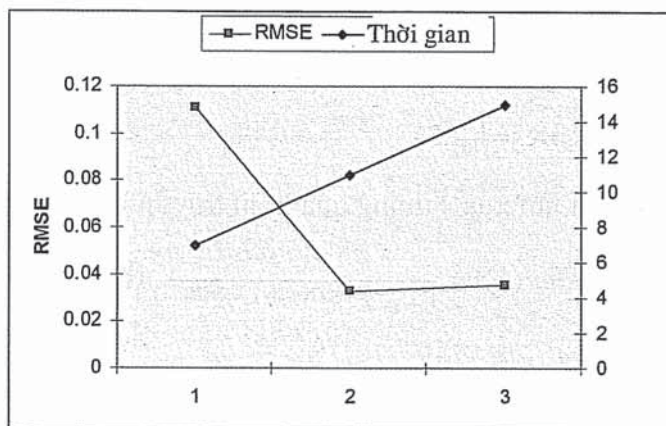
$w_{ij}(\text{mới}) = w_{ij}(\text{cũ}) + \alpha \cdot \Delta w_{ij}$

P_{im} : giá trị đầu vào của nút nhập thứ i đối với mẫu thứ m .

*** Bước 9:**

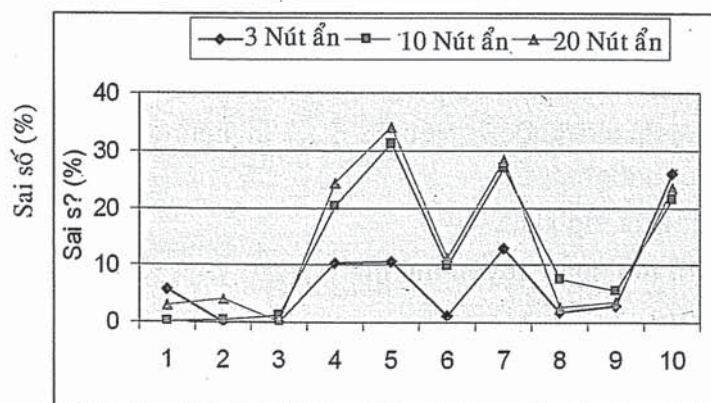
Trở lại bước 2:

Để xác định số lớp ẩn của mô hình mạng nơron, mạng nơron với 0, 1 và 2 lớp ẩn được sử dụng. Khi so sánh sai số trung bình bình phương để thấy rằng mạng có 1 lớp ẩn cho sai số nhỏ nhất và thời gian luyện mạng hợp lý.



Hình 1. Đồ thị biểu diễn sai số quân phương và thời gian luyện mạng theo số lớp ẩn

Tương tự, để tìm số nút ẩn tối ưu các mô hình mạng nơron với số nút ẩn khác nhau sẽ được luyện. Mạng nào cho % sai số nhỏ nhất sẽ được chọn.



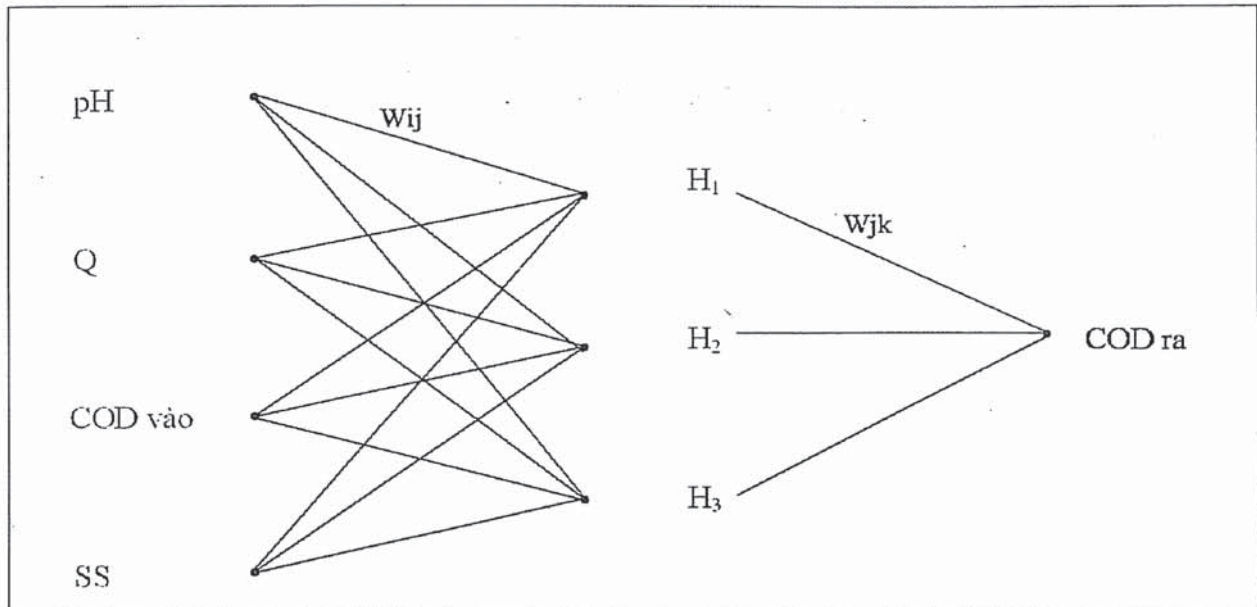
Hình 2. Đồ thị so sánh % sai số giữa các kết quả tính toán trong ba mạng nơron có số nút ẩn khác nhau.

Rõ ràng là khi mạng có số nút ẩn là 10 và 20 thì kết quả cho sai số lớn, còn với 3 nút ẩn chương trình cho kết quả chấp nhận được. Như vậy, mạng có 3 nút ẩn được chọn để sử dụng tính toán. Từ những kết quả trên ta có cấu trúc của mạng nơron trong ví dụ này như sau:

Biến nhập P_i
($i=1.2.3.4$)

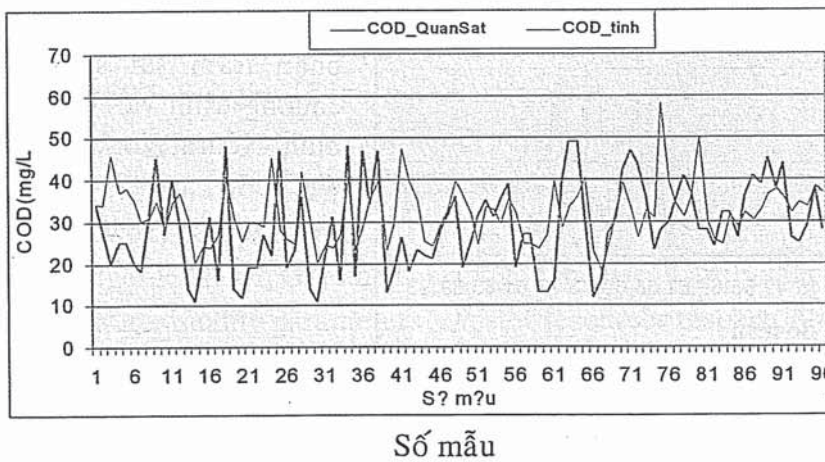
Lớp ẩn H_j
($j=1.2.3$)

Biến xuất Y_k
($k=1$)



Kết quả:

Kết quả tính toán giá trị COD thực tế và COD tính toán được thể hiện qua các đồ thị.

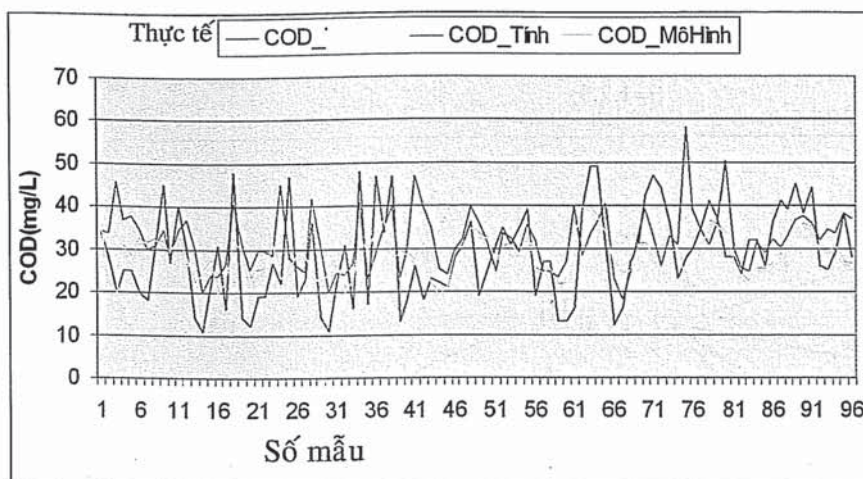


Hình 3. Đồ thị biểu diễn giá trị COD thực tế quan sát và COD tính toán của chương trình

Từ đồ thị trên ta thấy giữa giá trị dự đoán và giá trị quan sát có chênh lệch, nhìn chung những chênh lệch này không cao lắm. Tuy nhiên, vẫn có một số mẫu cho giá trị dự đoán lớn trong khi giá trị quan sát khá nhỏ. Vì vậy, để kiểm

chứng độ chính xác của chương trình, kết quả tính toán ở trên sẽ được so sánh với kết quả của mô hình NeuralWorks Predict phiên bản 3.0 (Mỹ). NeuralWorks Predict được sử dụng để dự đoán và dự báo trong nhiều lĩnh

vực khác nhau vì chương trình dễ sử dụng và cho kết quả nhanh. Đồ thị sau cho ta thấy sự so sánh giữa giá trị tính toán của chương trình và của mô hình NeuralWork Predict.

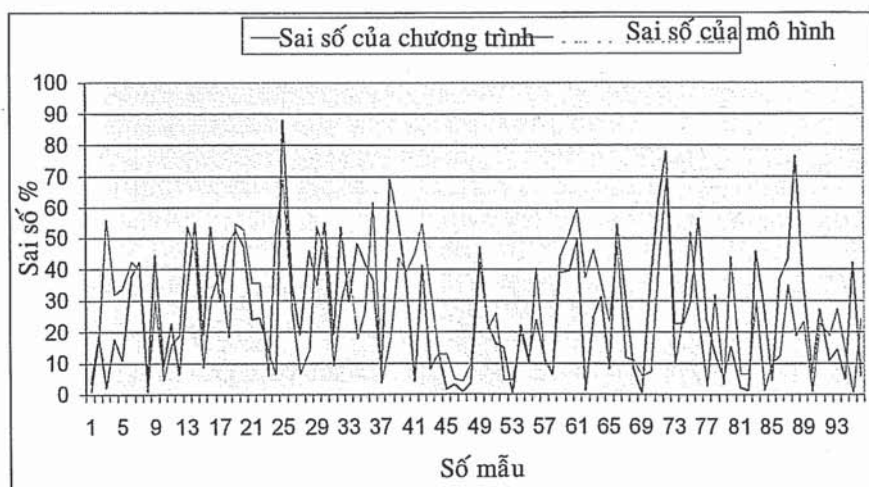


Hình 4. Đồ thị biểu diễn giá trị COD thực tế, COD tính toán và COD của mô hình NeuralWorks Predict.

Nhìn chung giá trị tính toán của chương trình sai biệt không lớn so với giá trị tính toán của mô hình NeuralWorks Predict. Để dễ kiểm

chứng ta so sánh % sai số giữa giá trị tính toán và giá trị quan sát.

$$SaiSo = \left| \frac{COD_{ThucTe} - COD_{tinh}}{COD_{tinh}} \right| * 100$$



Hình 5. Đồ thị biểu diễn phần trăm sai số của chương trình và của mô hình NeuralWorks Predict

Nhìn chung, sai số giữa chương trình và mô hình NeuralWorks Predict không có sự khác biệt nhiều, sai số trung bình của mô hình NeuralWorks Predict là 26,48% của chương trình là 27,14%. Như vậy, sai số trung bình của chương trình tương đối nhỏ,

kết quả tính toán của chương trình là chấp nhận được, có thể sử dụng để tính toán kết quả đầu ra của hệ thống xử lý nước thải.

4. Kết luận

Tính toán kết quả đầu ra của hệ thống xử lý nước thải là một vấn đề cần thiết cho

các hệ thống xử lý nước thải. Tùy theo quy mô của vấn đề, bài toán có thể dừng trong phạm vi một hệ thống xử lý nước thải hoặc có thể mở rộng cho nhiều hệ thống khác nhau. Dù đứng ở góc độ nào thì bài toán cũng góp phần giúp cho việc tính toán và dự

đoán các kết quả đầu ra của hệ thống sau khi xử lý nước thải được tiến hành nhanh chóng.

Ngoài ra, để biết được các thông số chất lượng nước sau khi xử lý người ta phải tiến hành làm thí nghiệm nhiều lần, rất tốn thời gian, tiền bạc và công sức. Như vậy, nếu có

một cơ sở dữ liệu hợp lý và đầy đủ thì chương trình tính toán kết quả đầu ra của hệ thống xử lý nước thải được xem như một phòng thí nghiệm ảo, ở đó ta có thể tính toán được các thông số chất lượng nước, kiểm soát được các quá trình xử lý của nhà máy và có phương pháp điều

chỉnh hợp lý mà không cần tiến hành lấy mẫu và làm thí nghiệm, tiết kiệm thời gian, tiền bạc và công sức. Đồng thời nếu trong quá trình xử lý có những thông số vượt quá giá trị giới hạn thì có thể kịp thời điều chỉnh để các thông số này đạt tiêu chuẩn xử lý.

Tài liệu tham khảo

1. Nguyễn Hoàng Phương, Bùi Công Cường, Nguyễn Doãn Phước, Phan Xuân Minh, Chu Văn Hỷ, Hệ mờ và ứng dụng. Nhà Xuất Bản Khoa học và Kỹ Thuật, năm 1998.
2. Nguyễn Đình Thúc, Trí tuệ nhân tạo - Mạng nơron phương pháp và ứng dụng. Nhà Xuất Bản Giáo Dục, năm 2000.
3. Lư Nhật Vinh. Kết hợp mạng nơron và logic mờ để giải quyết bài toán kinh tế, (2001).
4. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. "Artificial neural networks in Hydrology". Journal of Hydrologic Engineering. Vol 5 (No.2), 115 - 123, 2000.
5. Ben Krose, Patrick van der Smagt. An introduction to neural networks. 1996.
6. Christos Stergiou, Dimitrios Siganos, Neural networks.
7. Colin Fyfe. Artificial neural networks, 1996.
8. J.C. Chen, N.B. Chang. W.K. Shieh, Assessing wastewater reclamation potential by neural network model, (2003).
9. Holger R.Maier, Graeme C.Dandy. "The use of artificial neural networks for the prediction of water quality parameters", Water resources research, Vol. 32 (No.4), 1013 - 1022, 1996.
10. Holger R.Maier, Graeme C.Dandy. Neural networks for predictions and forecasting of water resources variables: a review of modelling issues and applications, 1999,
11. James A. Freeman, David M. Skapura, Neural Networks Algorithms, Applications, and Programming Techniques, Addison - Wesley Publishing Company.
12. P. S. Neelakanta; Dolores DeGroff, Neural Network Modeling: Statistical Mechanics and Cybernetic Perspectives, CRC Press. 1994.
13. M. Purvis, N. Kasabov, G. Benwell, Q. Zhou, F. Zhang, Neuro-fuzzy methods for environmental modelling.
14. Ruby L.Kenedy, Yuchun Lee, Benjamin Van Roy, Christopher D.Red, Dr.Richard P. Lippmann, Solving data mining problems through pattern Recognition, Prentice Hall PRT, 1995 - 1997.