Development and Implementation of a Corrosion Inhibitor Chatbot Using Bidirectional Long Short-Term Memory

Nibras Bahy Ardyansyah[1], Dzaki Asari Surya Putra[1], Nicholaus Verdhy Putranto[2], Gustina Alfa Trisnapradika[2], Muhammad Akrom[2]

[1] Informatics Engineering Program, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia

[2] Research Center for Quantum Computing and Materials Informatics, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia

Accepted 21 August 2024
Approved 14 March 2025

***Abstract*— This research delves into the intricate phenomenon of corrosion, a process entailing material degradation through chemical reactions with the environment, causing consequential losses across diverse sectors. In response, corrosion inhibitors are a proactive measure to counteract this deleterious impact. Despite their paramount significance, public awareness regarding corrosion and inhibitors remains limited; necessitating intensified educational efforts. The primary focus of this study is developing a Chatbot system designed to disseminate information on corrosion, inhibitors, and related topics. Employing the Machine Learning Life Cycle model, a deep learning approach, specifically the Bidirectional Long Short-Term Memory (BLSTM) architecture, is utilized to construct an optimized Chatbot model. Post-training evaluation of the BLSTM model reveals noteworthy performance metrics, including a remarkable 99% accuracy rate and a substantial 96% validation accuracy over 100 epochs. Training and validation losses are reported as 0.1112 and 0.4228, respectively. In conclusion, the BLSTM algorithm is an effective tool for training and enhancing Chatbot models, ensuring commendable corrosion awareness and inhibition performance.**

***Index Terms*— corrosion; inhibitor; Chatbot; deep learning; LSTM.**

I. INTRODUCTION

Corrosion is a process of material degradation that occurs through chemical reactions with the surrounding environment, causing structural damage and a decrease in material quality[1], [2]. Although it occurs frequently in everyday life, corrosion often goes unnoticed. Lack of awareness of corrosion symptoms can result in ignorance of the potential hazards that may arise due to structural damage and material deterioration. Environmental factors such as water, air, and certain chemicals play a role in accelerating this process[3]. Contact between two dissimilar metals in an electrolyte, microorganism interaction, as well as stress-induced corrosion can also accelerate material deterioration[4]. Corrosion not only impacts the degradation of materials and infrastructure but also has a wider impact, penetrating various interrelated sectors and playing an important role in modern life. including economic, environmental, and security industries[5], [6]. Corrosion prevention can be done through various methods that have been proven effective, such as metal surface coating, cathodic protection, and the use of corrosion inhibitors [7], [8]. The use of corrosion inhibitors is an important step that can inhibit the rate of corrosion reactions by adding certain chemical compounds into a corrosive environment[5], [9]. This method helps extend the material's service life and reduces maintenance and replacement costs caused by corrosion damage. However, general knowledge about corrosion and corrosion inhibitors is still limited. Educational efforts are needed to improve public understanding of corrosion. An effective information delivery method that is accessible to a wide range of people is key to raising awareness about corrosion prevention [10], [11].

Chatbot is a program designed to undergo human-computer interaction by utilizing Artificial Intelligence (AI) systems such as Natural Language Processing (NLP) [12], [13]. In addition, Chatbot can also be used as an effective tool to provide information related to technical issues such as corrosion. Implementing this Chatbot can create new opportunities to improve the accessibility of information regarding corrosion. Chatbots can accept various inputs, such as text, and provide responses based on pre-programmed patterns[14]. There are two main classifications in Chatbot, namely open domain and closed domain. Open-domain chatbots can respond appropriately to a wide range of general topics. In contrast, closed-domain chatbots can only provide responses related to specific topics and may not be effective in responding to other topics[15]. Chatbots have a high degree of flexibility, allowing for customization and training in multiple languages, thus meeting diverse and specific needs across different sectors. This capability allows chatbots to adapt to local languages, cultures, and user preferences, ensuring effective and relevant interactions across multiple contexts. As a result of this flexibility, chatbots can function optimally in a wide range of industries, including customer service, education, and healthcare, providing solutions tailored to the evolving needs of the market [16]. In developing a *Chatbot*, a model that can undergo training and testing processes using machine learning algorithms, such as Neural Networks, is needed [17]. This journal discusses the implementation or development of NLP systems, namely the corrosion inhibitor *Chatbot* using the *Deep Learning* approach and the *Bidirectional Long Short-Term Memory* (BLSTM) algorithm.

Some methods commonly applied in Chatbot development involve Recurrent Neural Network and Long Short-Term Memory (RNN-LSTM) [18], Bidirectional Long Short-Term Memory (BLSTM), and Natural Language Processing (NLP) [19]. With a structure designed to remember information, the LSTM (Long Short-Term Memory) algorithm eases the interaction between users and computers, allowing for more natural and effective conversations. Due to its ability to handle sequential data and recognize complex patterns in text. LSTM enables natural language processing that makes it easier for users to interact with computer [20]. Therefore, the LSTM method is becoming very popular in chatbot development. With the use of LSTM, a chatbot can receive input and generate output based on previously learned patterns [21].

This research brings innovation by applying the LSTM algorithm to Chatbot development. The main focus of the Chatbot is to provide responses related to user questions regarding information and knowledge about corrosion inhibitors and provide effective suggestions for corrosion prevention. The LSTM model training process was conducted independently using the Python platform and Jupyter Notebook. The strategic decision to choose LSTM as the main algorithm was based on the results of previous studies that showed superior classification accuracy compared to other methods, such as RNN and K-Kearnest Neighbors (KNN) [22]. Thus, this research proposes using LSTM as a more reliable foundation for building a Chatbot that can provide innovative solutions to questions and challenges surrounding corrosion inhibitors.

II. METHODOLOGY

This research was conducted through a series of machine learning model development processes. Figure 1 overviews this research's machine learning model development process. The development begins with collecting relevant data on corrosion and corrosion inhibitors and then exploring the data to understand its patterns and characteristics. Next, data preparation is performed, including data cleaning and transformation, to ensure its quality. Once the data is ready, a splitting stage is performed to divide the data into training and testing sets. A suitable algorithm is selected and trained using the training data in the modelling stage. Then, the model is evaluated using the testing data at the evaluation stage to measure the performance and accuracy of the model. Once the model is deemed adequate, the next step is requirement chatbot, where we test the chatbot model that has been developed to ensure that it meets the needs and then deployment, which is the implementation of the model into Streamlit to enable the model to be used effectively by users.

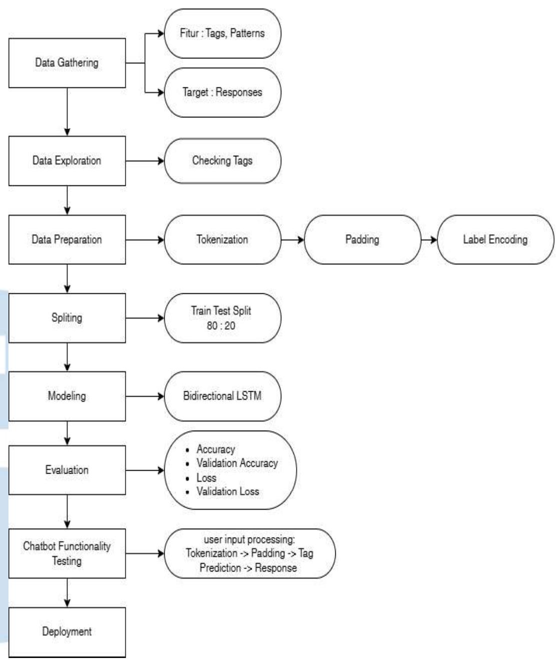

Fig. 1. ML model development Flowchart

A. Data Gathering

The initial stage of development involved data collection, which was conducted through research and the manual gathering of relevant data. This process was based on a literature review of corrosion and corrosion inhibitors from various reliable online sources. The choice of sources, such as Alodokter, HaloDoc, DrugBank, IDNmedis, Vinmec, PubChem, Drugs, Alomedika, Medicinka, and others, was driven by their relevance to the research topic. These platforms provide validated information on chemical substances, corrosion mechanisms, and inhibitors essential for building a high-quality dataset. The characteristics of the data were determined based on specific keywords related to corrosion and corrosion inhibitors. These keywords included corrosion, corrosion inhibitor, chemical substances, metal protection, and environmental impact. Data filtering used these keywords to ensure the collected information was relevant, accurate, and aligned with the research goals. This filtering process focused on extracting content that could serve as training patterns for the Chatbot. Additionally, the effort was made to include diverse questions and answers related to corrosion, ensuring comprehensive coverage of user queries.

Once the data was collected, it was organized into a dataset to train the chatbot model algorithm. The dataset was stored in JavaScript Object Notation (JSON) format and designed with the following structure:

- Tags: To group similar text data and use the same output as a target to train the neural network.
- Patterns: A component containing input pattern data expected to match the user input. Patterns are used as predictors.
- Responses: This is part of the data that contains answers or outputs that will be sent based on index tags and patterns determined by the system.

B. Data Exploration

The process at this stage serves as the first qualification before conducting model development. The data collected includes information related to corrosion, corrosion prevention, inhibitors, types of inhibitors, inhibitor mechanisms, corrosion inhibitors, drugs, expired drugs, expired drugs, corrosion inhibitors, and names of chemical compounds. This data will be used as answers or responses in the dataset. The developed chatbot uses a single-response approach, where each question from the user will be answered with one short and direct response according to the context of the question asked.

C. Data Preparation

In this stage, data processing is carried out to convert raw data into data ready to be used in the machine learning model training process at a later stage. Data exploration and data preparation, also known as data preprocessing, aims to ensure that the data used in Chatbot model training is clean, consistent, and representative so that it can produce better final results[22], [23]. The stages involved transforming the data into a suitable dataset format, using the panda's library to facilitate manipulation or modification, performing tokenization to convert the text in the pattern into a sequence of numbers, adding padding to the sequence of numbers to have the same length, and converting tag labels into numbers using the LabelEncoder technique. The data that has passed these stages has been optimally prepared for use in model training.

D. Splitting

The next stage is splitting the preprocessed dataset into two main parts: training and validation data. This division aims to train the model with some data and test its performance using data that the model has never seen before. The training process is performed for 100 epochs to improve the accuracy and performance of the model. The train test split function divides the dataset with a proportion of 80:20, where 80% becomes training data, and 20% becomes validation data. The training data is used as input and output for the model, while the validation data is used to evaluate the model's performance during training.

E. Modeling

The next stage in development involves the modelling process, which includes selecting an effective algorithm to recognize and respond to user queries accurately. The deep learning algorithm chosen for training the Chatbot is LSTM, which is known for achieving a high level of accuracy, superior response capabilities, and quick adaptability [24]. This study enhances the LSTM model by implementing a Bidirectional LSTM (BLSTM) structure. BLSTM processes sequential data in both forward and backward directions, making it particularly effective in understanding the context of user input. BLSTM was selected due to its ability to capture relationships between preceding and succeeding words in a sentence, which is crucial for NLP tasks such as Chatbot development. This capability allows the Chatbot to generate more accurate and context-aware responses than standard LSTM models that process data in only one direction. The model structure includes an Embedding layer, Bidirectional LSTM, Dropout layer, Normalization layer, Dense layer, and an additional LSTM layer. The model is compiled using the Adam optimizer and the categorical_crossentropy loss function, with accuracy as the primary evaluation metric. The model is trained using preprocessed data to maximize its performance in effectively recognizing and responding to user queries.

F. Evaluation

In this evaluation stage, the results of the trained model are displayed using an evaluation matrix. The evaluation metrics used include accuracy, validation accuracy, loss, and validation loss. In this model, the loss function applied is categorical cross-entropy, while the optimization uses the Adam method. Furthermore, to monitor the model's performance at each epoch, visualization is performed by displaying training accuracy, validation accuracy, training loss, and validation loss graphs. This evaluation process provides a holistic picture of the extent to which the model can provide accurate and consistent responses. The accuracy and loss graphs provide insight into the model's performance during the training and validation. After the evaluation process, the best-trained model can be selected for use in the Chatbot implementation[25].

G. Requirement Chatbot

At this stage, the best Chatbot model is used for the user input function, which aims to provide interaction between the user and the trained model. The user enters sentence patterns as input, and this function will process, perform class prediction, and display random responses from the Chatbot based on the pre-trained model. The preprocessing process involves several steps: character removal, conversion to lowercase, and tokenization. After getting the results from tokenization, the data is converted into a numeric sequence and padding is performed if needed. Next, the model performs tag label prediction, and the corresponding response is retrieved from the response dataset based on the tag label. The result of this interaction is to display the user input and a randomized response from the Chatbot.

H. Deployment

At this stage, the Chatbot model will be deployed for users to access online. In this research, the deployment process uses the Streamlit framework with the Python programming language. An environment with the library scikit-learn version 1.3.2, tensorflow 2.12.0, numpy 1.23, and Streamlit 1.29.0 is required to run the development process.

III. Result and Discussion

This research describes the development of a Chatbot to help improve public understanding of corrosion. This chatbot was developed using machine learning algorithms based on natural language processing (NLP) techniques. These methodologies enable the Chatbot to process natural language input effectively, providing accurate and contextually relevant responses to users' inquiries. The dataset used to train the Chatbot model algorithm is a manual dataset in the form of a JSON file. This dataset stores several power components, namely intents, tags, patterns, and responses, forming the foundation for the Chatbot's decision-making and response-generation capabilities. The structure of the dataset in Figure 2 allows the Chatbot to understand and provide appropriate responses to the questions asked by the user according to the specified topic. The model development involved using the collected dataset in JSON form and converting it into a data frame consisting of pattern and tag columns as the main columns. This process is done to organize the data to make it easier to process and structure. The data in the response column is used to provide answers that match the question based on the tags generated by the model, and they are randomly selected to increase the variety of responses. This dataset has 1386 rows representing diverse user inputs and is categorized into 298 unique tags. This extensive and well-curated dataset ensures the Chatbot's robustness and adaptability, allowing it to handle a broad spectrum of corrosion-related questions. By leveraging this comprehensive data, the Chatbot is better equipped to assist users, delivering clear and informative responses while enhancing the overall user experience.

Tokenization is applied to the pattern text, transforming it into a sequence of numerical representations. This step is critical in data preprocessing for natural language processing (NLP) models. Subsequently, padding is performed by adding zeros as prefixes or suffixes, standardizing the length of the numerical sequences across all data samples to maintain uniform input dimensions during the training process. Following this, the Label Encoding process is applied to the target variable. Specifically, the data in the tag column converts categorical labels into a numeric representation, typically in the form of binary vectors. The results of data preparation or preparation in x and y can be seen in Figure 3.

```
{"intents": [
  {
      "tag": "greetings",
      "patterns": [
        "hello",
        "Hi",
        "hello bot",
        "Hi bot",
        "Can you help me?"
      ],
      "responses": [
        "Hello! Can I help you?",
        "Hi! How can I help you?",
        "Hi! How are you? Anything to ask?",
        "Hello! How can I help you?",
        "Hi! How can I help you today?",
        "Hi there! How can I help you?",
        "Welcome! What can I do for you?"
      ]
  },
```

Fig. 2. JSON database

```
X shape =  (1387, 10)
y shape =  (1387,)
num of classes =  299

X

array([[339,   0,   0, ...,   0,   0,   0],
       [340,   0,   0, ...,   0,   0,   0],
       [339,  40,   0, ...,   0,   0,   0],
       ...,
       [  1,   2,   5, ...,   0,   0,   0],
       [  6,   4, 248, ...,   0,   0,   0],
       [  1,   2,   8, ...,   0,   0,   0]])

y

array([277, 277, 277, ..., 243, 243, 243])
```

Fig. 3. Data Preparation X dan Y

The Bidirectional LSTM deep learning algorithm is used to develop the Chatbot model on the data processed for the training process. In Figure 4, we can see the architectural structure of the model, which consists of several layers. The process starts with the input layer as the first layer, which receives a batch of sequences with a sequence length of 11. The second layer is the embedding layer, which receives the input from the previous layer and converts it into a vector with 100 dimensions. The third layer is a Bidirectional LSTM layer with 256 parameter units, which can generate a sequence of values in the input sequence and is equipped with a dropout to prevent overfitting. The dropout layer is further used to prevent overfitting and improve the capabilities of the Chatbot. The normalization layer in each layer is used to optimize the training process. Then, the second LSTM layer is equipped with dropout and normalization to prevent overfitting and normalize the data. After that, a dense layer with 64 units and a ReLU activation function is used to provide non-linearity to the model. The dropout and normalization layers are again applied before the softmax activation function is used. The LSTM model was trained for 100 iterations (epochs) to achieve optimal results. The model was compiled using the Adam optimizer and the categorical_crossentropy loss function, with the evaluation matrix being accuracy.

```
_________________________________________________
 Layer (type)                Output Shape
=================================================
 embedding (Embedding)       (None, 10, 100)

 bidirectional (Bidirection  (None, 10, 256)
 al)

 dropout (Dropout)           (None, 10, 256)

 layer_normalization (Layer  (None, 10, 256)
 Normalization)

 lstm_1 (LSTM)               (None, 128)

 dropout_1 (Dropout)         (None, 128)

 layer_normalization_1 (Lay  (None, 128)
 erNormalization)

 dense (Dense)               (None, 64)

 layer_normalization_2 (Lay  (None, 64)
 erNormalization)

 dropout_2 (Dropout)         (None, 64)

 dense_1 (Dense)             (None, 299)

=================================================
```

Fig. 4. LSTM Model Structure

The model training process starts by dividing the dataset into two main parts: training and validation data. This division aims to train the model using part of the data and test its performance using data that the model has never seen before. The training process is carried out for 100 epochs to improve the accuracy and performance of the model. In the training implementation, the train test split function divides the dataset with a proportion of 80:20, where 80% of the data becomes the training part and 20% becomes the validation part. The training data is used as input and output for the model, while the validation data is used to evaluate the model's performance during the training process. This process aims to allow the model to learn from the training data.

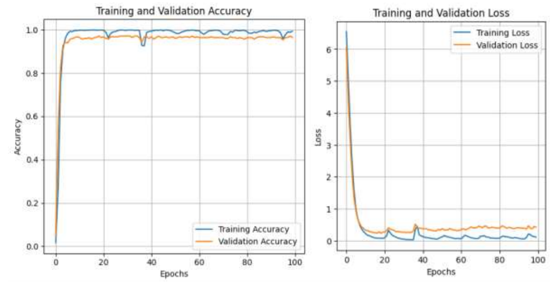

Fig. 5. Accuracy and loss graph of training and validation

TABLE I. ACCURACY AND LOSS METRICS, ALONG WITH VALIDATION OF LSTM TRAINING

Epoch	Traning Validation Result			
	Accuracy	*Val_Accuracy*	*Loss*	*Val_Loss*
10	0.9964	0.9676	0.2844	0.3861
20	0.9991	0.9640	0.0761	0.2735
30	0.9973	0.9712	0.0565	0.2676
40	0.9919	0.9604	0.1542	0.3889
50	0.9910	0.9676	0.1010	0.3320
60	0.9955	0.9676	0.0658	0.3302
70	0.9928	0.9640	0.0702	0.4597
80	0.9973	0.9568	0.0734	0.3977
90	0.9946	0.9640	0.0863	0.4090
100	0.9937	0.9640	0.1112	0.4228

Based on Table 1, after going through 100 epochs, the LSTM model achieved a high accuracy of 99% and a validation accuracy of 96.40% with a loss value of 0.0863 and a validation loss of 0.1112. These results show that the model performs excellently on training data and can predict well on validation data. This model's evaluation metrics include accuracy, validation accuracy, loss, and validation loss. The loss function applied to this model is sparse categorical cross-entropy, and the optimizer used is Adam. Model performance evaluation is done by visualizing the graphs of Training Accuracy, Validation Accuracy, Training Loss, and Validation Loss. In Figure 5, it can be seen that the training results show the ability of the model to learn the training data well. However, it should be noted that the validation accuracy is slightly lower than the training accuracy, which may indicate overfitting. Similarly, the loss graph shows that the validation loss reduces the training loss and validation loss values quickly, but the value fluctuates after a few epochs. Therefore, these results show that the model can understand the patterns in the data well and provide good predictions.

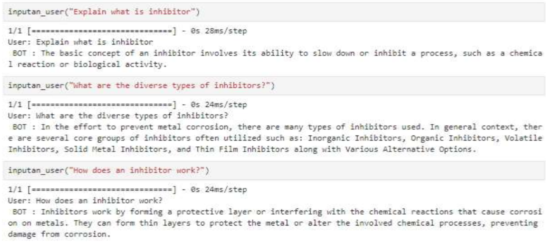

Fig. 6. Results of testing the chatbot

After preprocessing, the model predicts the tag label based on the input, and the corresponding response is taken from the pre-trained responses dataset. Figure 6 shows several test results by entering various questions, demonstrating that the Chatbot model can predict answers accurately. Unlike a Frequently Asked Questions (FAQ) page, which relies on static keyword-based search, this Chatbot uses a machine learning model based on Bidirectional Long Short-Term Memory (BLSTM). This allows the Chatbot to understand the context of user input and provide more dynamic, accurate, and context-aware responses. This capability makes the Chatbot a more effective and versatile tool for delivering information than traditional FAQ systems. Deployment is done so that users can use the Chatbot model created. The Chatbot model in this study was built into a web-based application using a framework or library in the Python programming language, Streamlit, to design the user interface. The results of the deployment stage can be seen in Figure 7, which shows the Chatbot web application interface. Users can enter a message or question into the column provided, and by pressing the send button, the Chatbot will provide an answer that matches the question. If the user wants to ask further questions, they can enter a new message and send it to the Chatbot again in the same way. Unlike previous studies that used LSTM models, this study implements Bidirectional LSTM (BLSTM). This innovation improves the model's ability to capture context in user queries, resulting in better accuracy and more relevant responses. This study outperforms previous research by achieving a higher validation accuracy of 96.40%, compared to 82.67% in earlier studies that used LSTM. Despite some indications of overfitting, the model remains capable of providing accurate and appropriate responses.

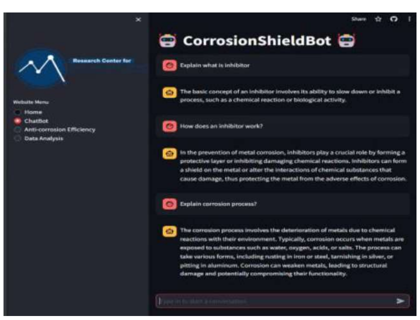

Fig. 7. Chatbot website interface

IV. Conclusion

Implementing the Long Short-Term Memory (LSTM) algorithm on the corrosion inhibitor Chatbot website has proven instrumental in enhancing the understanding of corrosion. The Chatbot training model, utilizing the LSTM algorithm, has demonstrated impressive performance, achieving an accuracy rate of 99% and a minimal loss value of 0.1112. Furthermore, integrating the Chatbot into a web-based application using the Streamlit framework has expanded its accessibility, allowing users to access it online and locally. The successful implementation of the Long Short-Term Memory (LSTM) algorithm on the corrosion inhibitor Chatbot website proved instrumental in improving the understanding of corrosion. The Chatbot training model, which utilizes the LSTM algorithm, has demonstrated commendable performance, achieving a remarkable accuracy rate of 99% and a minimal loss of 0.1112. Moreover, the successful integration of the Chatbot into a web-based application using the Streamlit framework or library has expanded its accessibility, allowing users to access it both online and locally. Designed to deliver information about corrosion, this Chatbot is an educational tool that is beneficial and relevant to a wide range of users. Nevertheless, it is important to recognize the limitations of this study's data. Future research efforts should prioritise testing and training on larger data sets to improve the reliability of the Chatbot and gain a more comprehensive understanding of the effectiveness and efficiency of the algorithms used. This approach ensures the continuous evolution and refinement of the Chatbot's capabilities, contributing to disseminating more accurate and robust corrosion information.

References

[1] C. A. P. Sumarjono, M. Akrom, and G. A. Trisnapradika, "Perbandingan Model Machine Learning Terbaik untuk Memprediksi Kemampuan Penghambatan Korosi oleh Senyawa Benzimidazole," *Techno.Com*, vol. 22, no. 4, Art. no. 4, Nov. 2023, doi: 10.33633/tc.v22i4.9201.

[2] M. Akrom, DFT Investigation of Syzygium Aromaticum and Nicotiana Tabacum Extracts as Corrosion Inhibitor, Science Tech: Jurnal Ilmu Pengetahuan dan Teknologi, Volume 8, Issue 1, Pages 42-48 (2022), https://doi.org/10.30738/st.vol8.no1.a11775.

[3] M. Akrom, T. Sutojo, Investigasi Model Machine Learning Berbasis QSPR pada Inhibitor Korosi Pirimidin Investigation of QSPR-Based Machine Learning Models in Pyrimidine Corrosion Inhibitors, Eksergi, 20(2), (2023), https://doi.org/10.31315/e.v20i2.9864.

[4] M. Akrom, "INVESTIGATION OF NATURAL EXTRACTS AS GREEN CORROSION INHIBITORS IN STEEL USING DENSITY FUNCTIONAL THEORY," *Jurnal Teori dan Aplikasi Fisika*, vol. 10, no. 1, Art. no. 1, Jan. 2022.

[5] M. Akrom *et al.*, "Artificial Intelligence Berbasis QSPR Dalam Kajian Inhibitor Korosi," *JoMMiT : Jurnal Multi Media dan IT*, vol. 7, no. 1, pp. 015–020, Jul. 2023, doi: 10.46961/jommit.v7i1.721.

[6] M. Akrom, Green corrosion inhibitors for iron alloys: a comprehensive review of integrating data-driven forecasting, density functional theory simulations, and experimental investigation, Journal of Multiscale Materials

Informatics, Volume 1, Issue 1, Pages 22-37 (2024), https://doi.org/10.62411/jimat.v1i1.10495.

[7] M. Akrom, S. Rustad, H.K. Dipojono, R. Maezono, A comprehensive approach utilizing quantum machine learning in the study of corrosion inhibition on quinoxaline compounds, Artificial Intelligence Chemistry, Volume 2, Issue 2, Pages 100073 (2024), https://doi.org/10.1016/j.aichem.2024.100073.

[8] N. V. Putranto, M. Akrom, and G. A. Trinapradika, "Implementasi Fungsi Polinomial pada Algoritma Gradient Boosting Regressor: Studi Regresi pada Dataset Obat-Obatan Kadaluarsa Sebagai Material Antikorosi," *Jurnal Teknologi dan Manajemen Informatika*, vol. 9, no. 2, Art. no. 2, Dec. 2023, doi: 10.26905/jtmi.v9i2.11192.

[9] S. Marzorati, L. Verotta, and S. Trasatti, "Green Corrosion Inhibitors from Natural Sources and Biomass Wastes," *Molecules*, vol. 24, no. 1, p. 48, Dec. 2018, doi: 10.3390/molecules24010048.

[10] Z. M. Hanif, "Pengembangan Aplikasi Whatsapp Chatbot Untuk Pelayanan Akademik Di Perguruan Tinggi," Dec. 2021, Accessed: Jan. 09, 2024. [Online]. Available: https://dspace.uii.ac.id/handle/123456789/37445

[11] R. A. Yunmar and I. W. W. Wisesa, "Pengembangan Mobile based Question Answering System dengan Basis Pengetahuan Ontologi," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 4, Art. no. 4, Aug. 2020, doi: 10.25126/jtiik.2020742255.

[12] H. A. F. Muhyidin and L. Venica, "Pengembangan Chatbot untuk Meningkatkan Pengetahuan dan Kesadaran Keamanan Siber Menggunakan Long Short-Term Memory," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 5, no. 2, pp. 152–161, Oct. 2023, doi: 10.36499/jinrpl.v5i2.8818.

[13] E. L. Amalia and D. W. Wibowo, "Rancang Bangun Chatbot Untuk Meningkatkan Performa Bisnis," *jitika*, vol. 13, no. 2, p. 137, Oct. 2019, doi: 10.32815/jitika.v13i2.410.

[14] S. Rustad, M. Akrom, T. Sutojo, H.K. Dipojono, A feature restoration for machine learning on anti-corrosion materials, Case Studies in Chemical and Environmental Engineering, Volume 10, Pages 100902 (2024), https://doi.org/10.1016/j.cscee.2024.100902.

[15] K. Nimavat and T. Champaneria, "Chatbots: An overview. Types, Architecture, Tools and Future Possibilities," Oct. 2017.

[16] A. Elcholiqi and A. Musdholifah, "Chatbot in Bahasa Indonesia using NLP to Provide Banking Information," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 1, Art. no. 1, Jan. 2020, doi: 10.22146/ijccs.41289.

[17] R. Mahendra and M. Kamayani, "Menerapkan Algoritma Neural Network Pada Chatbot Mengenai Pariwisata Di Provinsi Bangka Belitung," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 7, no. 2, Art. no. 2, Sep. 2023, doi: 10.30645/j-sakti.v7i2.678.

[18] P. Anki, A. Bustamam, H. S. Al-Ash, and D. Sarwinda, "Intelligent Chatbot Adapted from Question and Answer System Using RNN-LSTM Model," *J. Phys.: Conf. Ser.*, vol. 1844, no. 1, p. 012001, Mar. 2021, doi: 10.1088/1742-6596/1844/1/012001.

[19] V. R. Prasetyo, N. Benarkah, and V. J. Chrisintha, "Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya," *Teknika*, vol. 10, no. 2, Art. no. 2, Jul. 2021, doi: 10.34148/teknika.v10i2.370.

[20] P. B. Wintoro, H. Hermawan, M. A. Muda, and Y. Mulyani, "Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 12, no. 1, p. 68, Jun. 2022, doi: 10.36448/expert.v12i1.2593.

[21] 1815061012 Hilmi Hermawan, "IMPLEMENTASI LONG SHORT-TERM MEMORY (LSTM) PADA CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS LAMPUNG." Accessed: Jan. 09, 2024. [Online]. Available: https://digilib.unila.ac.id/65316/

[22] A. Silvanie and R. Subekti, "APLIKASI CHATBOT UNTUK FAQ AKADEMIK DI IBI-K57 DENGAN LSTM DAN PENYEMATAN KATA," *JIKO (Jurnal Informatika dan Komputer)*, vol. 5, no. 1, Art. no. 1, Apr. 2022, doi: 10.33387/jiko.v5i1.3703.

[23] K. A. Nugraha and D. Sebastian, "Chatbot Layanan Akademik Menggunakan K-Nearest Neighbor," *JSI*, vol. 7, no. 1, pp. 11–19, Mar. 2021, doi: 10.34128/jsi.v7i1.285.

[24] Y. Denny, H. L. H. Spits Warnars, W. Budiharto, A. I. Kistijantoro, Y. Heryadi, and L. Lukas, "Lstm And Simple Rnn Comparison In The Problem Of Sequence To Sequence On Conversation Data Using Bahasa Indonesia," Sep. 2018, pp. 51–56. doi: 10.1109/INAPR.2018.8627029.

[25] F. Zakariya, J. Zeniarja, and S. Winarno, "Pengembangan Chatbot Kesehatan Mental Menggunakan Algoritma Long Short-Term Memory," vol. 8, 2024.