



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 8, August 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.524

 9940 572 462

 6381 907 438

 ijirset@gmail.com

 www.ijirset.com

Development of Voice User Interfaced Automated Storage and Retrieval System for Warehouse

Mayur Manohar Mhamunkar¹, Prof. James Curry²

Former MS Student, Industrial Engineering, Lamar University, Beaumont, Texas, USA¹

Department Chair and Associate Professor, Department of Industrial and Systems Engineering, Lamar University,
Beaumont, Texas, USA²

ABSTRACT: Background: To develop a voice-operated system for Automated Storage and Retrieval (ASR) in industrial settings, focusing on natural language processing (NLP) and deep learning integration. Methods: A Deep Neural Network (DNN) model was trained using a manually generated and algorithmically expanded dataset in JSON format. The data was pre-processed using tokenization, stemming, and a Bag of Words approach. The model architecture included an input layer with 164 neurons, two hidden layers, and an output layer with 32 neurons. The Softmax activation function was used for classification. The model underwent 1000 epochs of training, with 85% of the data for training and 15% for validation. Voice recognition was implemented using the Python Speech Recognition package, and responses were generated using Google Text-to-Speech. Findings: The DNN model achieved a training accuracy of 99.21% and a validation accuracy of 91.30%, with a system-wide accuracy of 83%. The system demonstrated reliable command recognition, with a 10% error rate and a 3% non-response rate. These results indicate effective NLP application in ASR systems, with potential overfitting observed between training and validation accuracies. The system's real-world performance in warehouse environments is promising, aligning with existing studies while contributing novel insights into domain-specific voice systems. Novelty and applications: Introduced a domain-specific voice system for ASR using advanced DNN and NLP techniques, addressing gaps in industrial applications.

KEYWORDS: ASR, Natural language processing (NLP), Machine learning, Neural Network, Text-to-speech, Deep learning.

I. INTRODUCTION

The Paper discusses the creation of a voice system for Automated Storage and Retrieval (ASR) systems, intended for use in warehouse and smaller storage environments. The project focuses on several important areas:

1. The effectiveness of natural language algorithms within industrial settings.
2. The efficacy of training set development procedures tailored for industrial applications.
3. The performance of standard deep learning techniques in an industrial context.
4. The overall system effectiveness.

Despite the widespread popularity of general-purpose voice assistants in the consumer market, there remains a significant gap in the deployment of domain-specific voice systems that require higher precision, operate with smaller training sets, and address narrower problem areas (Mennicken et al., 2018) [1]. This paper addresses this gap by presenting a case study on the creation of a domain-specific assistant, encompassing the development of training sets, algorithm creation, and system testing.

The paper presents a literature review, which examines previous work in robotics with human interfaces, voice response systems, outlines the development of the algorithms used in this project and discusses the results of the algorithm testing and system implementation. It concludes the report as well as summarizing the findings and implications of the research.

II. RELATED WORK

In signal processing (Gutkin et al., 2011), image processing (Karabatak et al., 2008), and information retrieval (Minghui et al., 2017), neural networks are extensively utilized [2,3,4]. Neural network architectures were created by Shang (2015) and Sordoni et al. (2015) for response generation [5,6]. Singh et al. (2018) developed chatbots

with TensorFlow by utilizing feedback systems and natural language processing [7]. This research informs command recognition for certain domains in ASR systems, which is akin to chatbot technology.

Sharma et al. (2017) investigated the development of chatbots using AI Markup Language, from Eliza to Alice. Dutta et al. (2007) and Lauria et al. (2002) investigated the use of NLP in robotics and education [9,10]. NLP was used in academic advising by Ghose (2013) and customer interaction by Prasomphan (2019), NLP models were created by Setiaji & Wibowo (2016) to measure the similarity of sentences [11,12,13]. These works demonstrate how NLP might improve interactions between humans and machines.

Voice command systems have been integrated with robotics, as seen in projects by Teller et al. (2010) and Correa et al. (2010) with forklifts, and Delden (2012) and Pires (2005) with pick-and-place tasks [14,15,16,17]. Bradshaw (2017) and Pease (2017) showed practical applications in warehouses [18, 19]. This project’s ASR system leverages these advancements to use natural language for intuitive human-robot interactions.

Despite the success of general-purpose voice assistants like Siri and Alexa, there is a gap in domain-specific systems for industrial environments. This project addresses this gap by developing a voice-operated ASR system, aiming to enhance efficiency and user interaction in storage and retrieval processes using advanced neural networks and NLP techniques.

III. METHODOLOGY

This section consists of the detailed steps of developing the algorithm for building the assistant to drive the AS/RS system. The basic steps of the algorithm are: 1. Prepared and generated the data in a question-and-answer structure in a JSON file format by using an algorithm developed for data generation. 2. Preprocessing of the data with NLTK (Lauria et al. .2002) library using Tokenization, Stemming and Bag of Word Approach. [10] 3. Neural network from TensorFlow library applied (Singh et al. 2018) [7]. 4. Dataset Training. 5. Accuracy of the training calculated. 6. Performed the test on the whole system and measured the accuracy and precision of the job performed by the system. A detailed flow chart for methodology of the proposed system Algorithm is shown in [Figure 1].

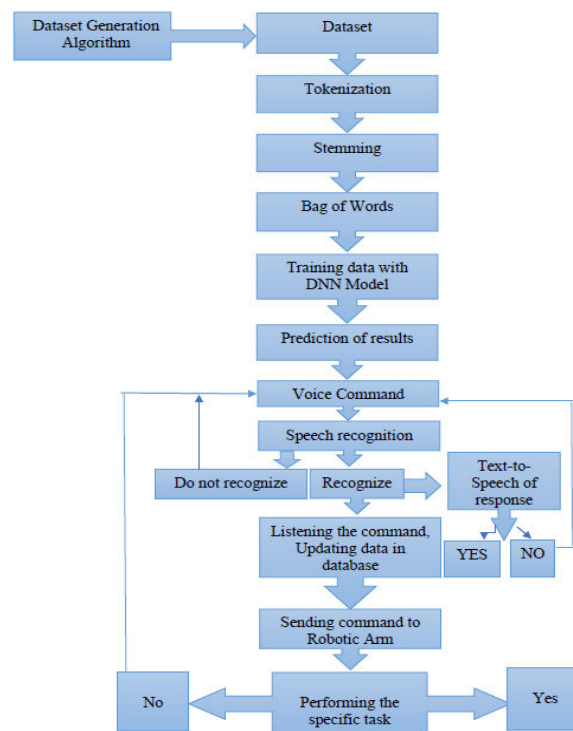


Figure 1. Flow chart for methodology of Proposed Algorithm

3.1 Dataset

The dataset for training the Deep Neural Network (DNN) model was generated partially manually and through an algorithm as shown in [Table 1]. Manually created sentences were expanded using synonyms to create a diverse set of similar sentences. The dataset, formatted in JSON, includes “Intent” (or “tag”), which specifies the user command, “Pattern,” which is the actual user command, and “Responses,” which are the system’s replies. For example, for the intent “greet,” commands like “Hi,” “Hello,” “Howdy,” etc., correspond to responses like “Hello!” “How may I help you?” etc. This dataset structure enables the model to learn patterns and provide accurate responses. A combination of handwritten and algorithm-generated data ensures comprehensive coverage of possible commands. Table 1 shows Dataset example is an as follows.

Table 1. Example of Dataset for the training of the DNN Model

Sr.No	Tag	Patterns	Responses
1	“greeting”	["Hi", "How are you", "howdy", "Is anyone there?", "Hello", "Good day", "Whats up", "Aloha", "how do you do", "hello", "hullo", "Hawai", "HI", "hi", "Hawaii"]	["Hello!", "Good to see you again!", "Hi there, how can I help?", "Aloha", "how do you do", "hello", "hullo", "Hawai", "HI", "hi", "howdy", "Hawaii"]

3.2 Natural Language Processing

The Natural Language Toolkit (NLTK) library (Sphinx, 2020) is use for preprocessing the dataset before training, it also offers various tools like “Word Tokenizer,” “n-gram,” “Lemmatizer,” and “Part-of-Speech Tagger (POS Tagger) [20].” The preprocessing steps included tokenization where sentences were split into individual words and stemming which reduced words to their root forms to minimize unique words, improve training efficiency.

3.2.1 Tokenization

Tokenization involves splitting sentences into individual words or tokens for further processing and tokens are separated by spaces, commas (,) or other special characters [20]. This process is essential for preparing the dataset for stemming and POS tagging. [Table 2] shows the samples before the tokenization process and samples after the tokenization process.

Table 2. Sample for Tokenized Data

Sr. No	Sentences before tokenization	Sentences after tokenization
1	["Hi", "How are you", "Is anyone there?", "Hello", "Good day", "What's up", "Aloha", "how do you do", "hello", "hullo", "Hawai", "HI", "hi", "howdy", "Hawaii"]	['Hi'] ['How', 'are', 'you'] ['Is', 'anyone', 'there', '?'] ['Hello'] ['Good', 'day'] ['What's', 'up'] ['Aloha'] ['how', 'do', 'you', 'do'] ['hello'] ['hullo'] ['Hawai'] ['HI'] ['hi'] ['howdy'] ['Hawaii']

3.2.2 Stemming

Stemming reduces tokens to their root forms which decreases the number of unique words in the dataset and enhances training efficiency. The Lancaster Stemmer algorithm was used for its precision and ability to exclude stop words, making it more suitable than other algorithms like Porter or Snowball [20]. [Table 3] shows the samples before the stemming process and samples after the stemming process.

Table 3. Stemming of Dataset

Sr.No	Words before stemming process	Words after the stemming process
1	['Hi'] ['How', 'are', 'you'] ['Is', 'anyone', 'there', '?'] ['Hello'] ['Good', 'day'] ['Whats', 'up'] ['Aloha'] ['how', 'do', 'you', 'do'] ['hello'] ['hullo'] ['Hawai'] ['HI'] ['hi'] ['howdy'] ['Hawaii'] ['cya'] ['See', 'you', 'later'] ['Goodbye']	['hi', 'how', 'ar', 'you', 'is', 'anyon', 'ther', 'hello', 'good', 'day', 'what', 'up', 'aloh', 'how', 'do', 'you', 'do', 'hello', 'hullo', 'hawa', 'hi', 'hi', 'howdy', 'hawai', 'cya', 'see', 'you', 'lat', 'goodby']

3.3 Deep Natural Language Processing

The process of translating human language into a format that the DNN model can comprehend is known as deep natural language processing or DNLP. The "Bag of Words" method was applied, in which each word's frequency in the phrases is tallied and shown as a binary representation (0s and 1s). The DNN model is then trained using this binary representation.

3.3.1 Bag of Words

The Bag of Words approach involves creating a frequency list of words from the dataset sentences. A for loop counts each word's occurrence, resulting in a binary list indicating the presence (1) or absence (0) of each word [20]. This binary data is structured into an array matrix using NumPy for training the model. Example of output matrix for application of 'Bag of Words' techniques is shown in [Table 4].

Table 4. Example of output matrix for application of 'Bag of Words' techniques

	Number of words								
Example No. of Sentences (commands) pattern	Me		AMay	I	Have	Get	Bring	coke	...so, on 164 words
	1		10	0	0	0	1	1	..
	1	1	0	0	0	1	0	1	...
	0		11	1	1	0	0	1	...
613 sentences.	

3.4 Deep Neural Network Design

As shown in [Figure 2] the input, hidden, and output layers make up the DNN model's three layers, X1, X2, X3.... Xn are the Input values (Independent Variables) to the network. W1, W2.... Wn are the calculated Weights of the network before feeding to the different hidden layers and output value of the network is Y. The binary word representation is represented by the 164 neurons that make up the input layer, Neurons in hidden layers process inputs by means of weighted sums and the processing of the hidden layers is used by the output layer to generate results.

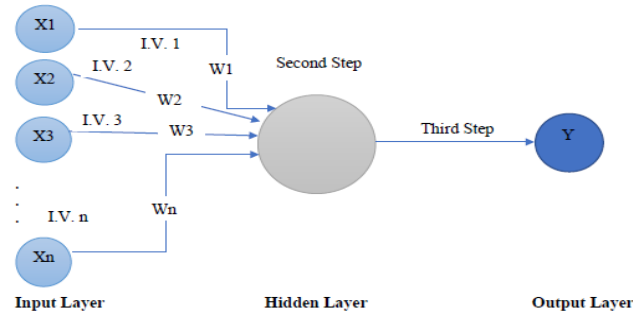


Figure 2. Basic Structure of Neural Network

3.4.1 Independent Variables

Independent variables in the DNN model are the binary numbers (0s and 1s) which represent words from the dataset and each sentence is converted into a binary format with 164 neurons in the input layer, one for each word.

3.4.2 Dependent Variables

The categorical outputs relating to the dataset intents are known as dependent variables. The outcome is based on the category with the highest probability among the 32 possible intentions that the model classifies commands into.

3.4.3 Activation Function

The Softmax activation function, applied at the output layer, converts logits into probabilities summing to one. This method is preferred over functions like ReLU for its accuracy in producing probabilities [21]. [Figure 3] shows the mathematical equations input and output effects of Softmax activation function used in the Neural network model in which y is the input value and p are probabilities of the output. The maximum probability was applied to the final model.

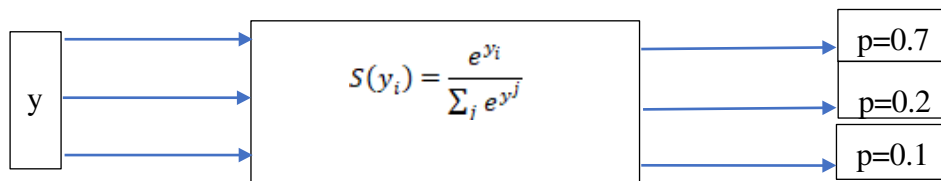


Figure 3. Softmax Activation Function

3.4.4 Gradient Descent

This process minimizes the cost function (C) produced by the difference between the actual value of the result and the value obtained by the output layer after feeding in the training stage of the (Tensorflow DNN library, 2018). The cost function, which calculates the difference between the actual and expected output values, is minimized by gradient descent. Over several training epochs, this iterative approach modifies neuron weights to increase model accuracy. [Figure 4] shows the Gradient descent Process, as shown $X1$ is an input parameter Y^{\wedge} is the output of the process and Y is an actual output of the DNN network model.

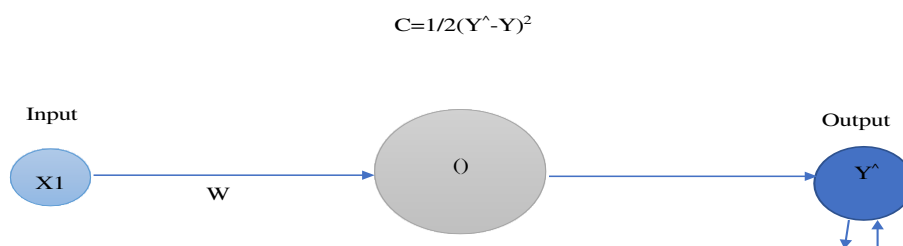


Figure 4. Stochastic Gradient Descent Process

3.4.5 Architecture of DNN

The newly developed DNN model architecture consists of an input layer with 164 neurons, an output layer with 32 neurons, two hidden layers with eight neurons each, and 1000 training epochs with a batch size of 8 for the training [22]. Of the dataset, 85% is the training set, while the remaining 15% is set aside for validation. As shown in Design architecture of DNN model in [Figure 5] X1, X2, X3...X164 are the input variables for the Input Layer from the users and Y1, Y2.... Y32 are the output from the DNN network of different probability values.

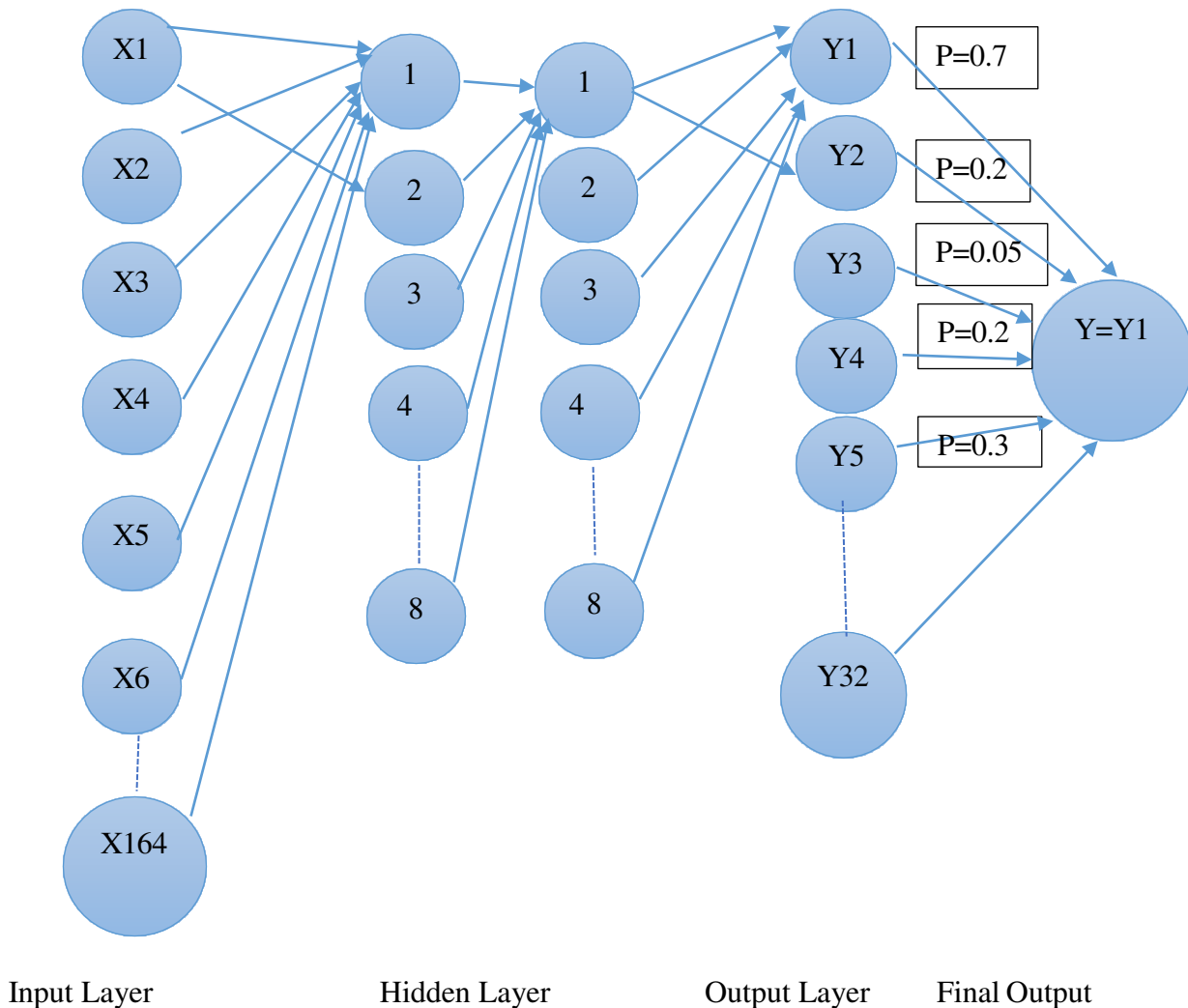


Figure 5. Design architecture of Deep Neural Network.

3.5 Data Storage and Retrieval

A function using Configure parser library and '.init' file format manages data storage and retrieval commands. It checks space availability for storing items and records storage, retrieval activities by ensuring efficient command execution and tracking. Once a command is given to store, it will retrieve the item on the shelf, the command will confirm the availability of the prescribed space for the item in a '.init' file. If the location is empty the function sends a command to the 'robot_Action' function, and the robot can do the specified job.

3.6 Voice Recognition and Text to Speech

Voice command recognition was developed by utilizing the Speech Recognition package in Python to translate spoken commands into text for the DNN model (speech Recognition library, 2019), the Google Text-to-Speech (gTTS)

module made it easier to translate text responses back into voice by allowing users to engage with the system naturally [23].

IV. EXPERIMENTAL RESULTS

This section presents the training, validation, and testing results for the DNN model. The model was trained on 613 sentences with an 85/15 split for training and validation. The outcomes show the model's intent prediction accuracy and overall system performance.

4.1 Model Training and validation

As per epochs shown in [Table 5] and line graph for Training and validation accuracy in [Figure 6] the model achieved 99.21% training accuracy with a loss of 0.04463 at the 1000th epoch, and 91.30% validation accuracy with a loss of 1.05875. The cross-validation technique has been used to find out if the model is overfitting or underfitting. The data was split into two parts – the training set and the validation set before the training process. This validation set has used to evaluate the performance of the model. Training accuracy was consistently higher than validation accuracy, indicating potential overfitting, with the final model achieving 91.30% accuracy for new data.

Table 5. Model Training Results

Epochs Number	Loss	Validation Loss	Accuracy	Validation Accuracy	Total Loss
600	0.00073	0.75813	99.94%	92.39%	0.00068
650	0.0014	0.76718	99.92%	92.39%	0.00126
700	0.854	0.89006	93.56%	91.30%	1.60778
750	0.00101	0.90328	99.92%	92.39%	0.00091
800	0.05133	0.98887	99.59%	90.22%	0.04625
850	0.00158	1.03518	99.90%	91.30%	0.00144
900	0.00461	1.06248	99.68%	91.30%	0.00417
950	0.0001	1.01509	100.00%	92.39%	0.0001
1000	0.04463	1.05875	99.21%	91.30%	0.04463

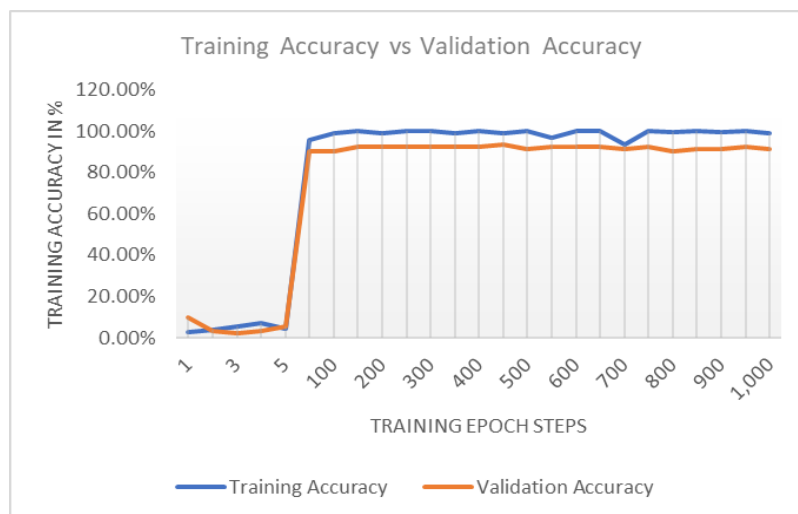


Figure 6. Line Chart on Training Accuracy Vs Validation Accuracy

4.2 Training and Validation Loss

This method was performed to validate the training process between some intervals. If the training loss is consistently lower than validation loss, which confirms underfitting, if training loss is higher, that validation loss is called overfitting. The above model is a slightly underfit model. As per [Figure 7], the training loss is lesser than the validation loss after 50 epochs. though both converged near the 700th epoch.

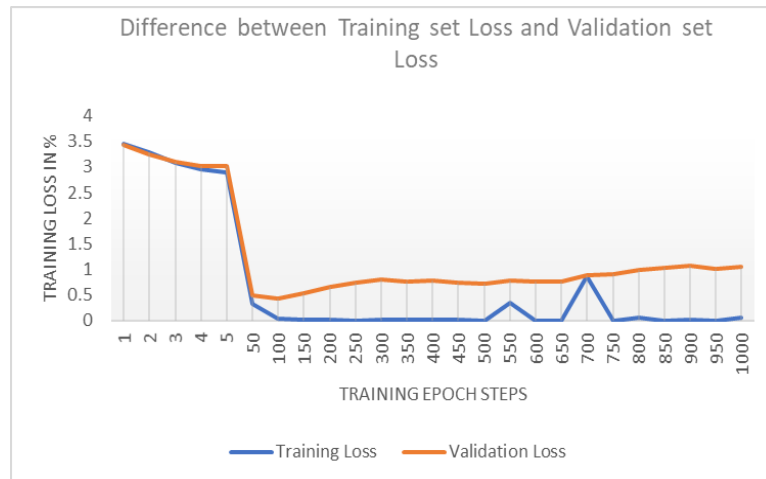


Figure 7. Line Graph Chart on Training Loss Vs Validation Loss

4.3 Model and System Accuracy

Random command testing (e.g., "Hi") showed accurate intent prediction (e.g., "greetings"), demonstrating the model's effectiveness. This decided intent will be then fed to the function to produce the output response, and this is possible because the function will take out the max float from the list. The overall accuracy of the system is measured by asking the questions. These questions are closely similar to the questions in the patterns of the dataset. Whenever the command or question is asked the system replies according to its understanding. This command takes out data from the robotAction.py File about the database of the system, and responds accordingly, also it will trigger robot action. Overall system accuracy was 83%, with a 10% rate of incorrect actions and a 3% rate of no response, highlighting reliable performance in real-world scenarios.

V. CONCLUSION

This research aimed to develop a system that communicates with humans and performs tasks in industrial and household environments with high accuracy, reducing human workload. The DNN model achieved 99.21% training accuracy and 91.30% validation accuracy, close to the targets of 99.99% and 95%, respectively. The system's overall accuracy was 83%, with a 10% error rate and a 3% non-response rate, demonstrating acceptable performance.

Future research can enhance the system by:

1. Implementing image processing to identify items or products.
2. Using convolutional neural networks to reduce reliance on voice commands.
3. Integrating a vision system with automatic entity detection.
4. Training the system with extensive image data to improve task performance and accuracy. These improvements could significantly increase system accuracy, expand its range of tasks, and further reduce human workload in various environments.

REFERENCES

1. Mennicken, Sarah, Brillman, Ruth, Cramer, Henriette, Thom, Jennifer., "Challenges and Methods in Design of Domain-Specific Voice Assistants." The 2018 AAI Spring Symposium Series. https://cdn.aai.org/Symposia/Spring/2018/SS-2018_Technical_Report_SS-18.pdf

2. Gutkin, R., Curtis, P.T., Green, C.J., Pinho, S.T., Robinson, P., Vangrattanachai, S., “On acoustic emission for failure investigation in CFRP: Pattern recognition and peak frequency analyses.” *Journal of Mechanical Systems and Signal Processing* 25 (2011) 1393–1407. <https://www.sciencedirect.com/science/article/abs/pii/S0888327010004176>
3. Karabatak, M., Cevdet, Ince M. “An expert system for detection of breast cancer based on association rules and neural network.” *Journal of Expert Systems with Applications* 36 (2009) 3465–3469. <https://www.sciencedirect.com/science/article/abs/pii/S0957417408001103>
4. Minghui, Q., Chen, Haiqing, Chen, Yan, Chu, Wei, Gao, Xing, Huang, Jun, Li, Feng-Lin, Wang, Siyu, Zhao, Weipeng. “AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine.” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, (2017) 498–503. https://www.researchgate.net/publication/318741788_AliMe_Chat_A_Sequence_to_Sequence_and_Rerank_base_d_Chatbot_Engine
5. Shang, Lifeng, Li, Hang, Lu, Zhengdong. “Neural Responding Machine for Short-Text Conversation.” *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.*, (2015): 1577–1586. <https://aclanthology.org/P15-1152.pdf>
6. Sordoni, Alessandro, Auli, Michael, Brockett, Chris, Dolan, Bill, Galley, Michel, Gao, Jianfeng, Ji, Yangfeng, Mitchell, Margaret, Nie, Jian-Yun. “A Neural Network Approach to Context-Sensitive Generation of Conversational Responses.” *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, (2015): 196–205. <https://aclanthology.org/N15-1020.pdf>
7. Singh, R., Mishra, Nitin, Paste, Manmath, Patel, Harshkumar, Shinde, Nirmala. “Chatbot using TensorFlow for small Businesses.” *Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018)* ISBN:978-1-5386-1974-2.
8. Sharma, Vibhor, Goyal, Monika, Malik, Drishti. “An Intelligent Behaviour Shown by Chatbot System.” *International Journal of New Technology and Research* Volume-3, Issue-4, (2017) 52-54. <https://media.neliti.com/media/publications/263312-an-intelligent-behaviour-shown-by-chatbo-7020467d.pdf>
9. Dutta, Debasatwa. “Developing an Intelligent Chat-bot Tool to assist high school students for learning general knowledge subjects.” *College of Computing Technical Reports* (2007): 01-12. <https://www.semanticscholar.org/paper/Developing-an-Intelligent-Chat-bot-Tool-to-Assist-Dutta/efca19604c4b6b9a05ac6838e3481a9aa027a210>
10. Lauria, Stanisla, Bugmanna, Guido, Klein Ewan, Kyriacou Theocharis. “Mobile robot programming using natural language.” *Journal Robotics and Autonomous Systems*, 38 (2002): 171–181.
11. Ghose, Supratip, Barua, Jagat Joyti. “Toward the implementation of a Topic specific Dialogue based Natural Language Chatbot as an Undergraduate Advisor.” *IEEE 2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, (2013): 1-5.
12. Prasomphan, Sathit. “Improvement of Chatbot in Trading System for SMEs by Using Deep Neural Network.” *IEEE 4th International Conference on Cloud Computing and Big Data Analytics*, (2019): 517–522. <https://www.semanticscholar.org/paper/Improvement-of-Chatbot-in-Trading-System-for-SMEs-Prasomphan/9cfb2ccd1bccfc4092f4b1cf9616fd8912f45e01#extracted>
13. Setiaji, Bayu, Wibowo, Ferry Wahyu. “Chatbot Using A Knowledge in Database.” *7th International Conference on Intelligent Systems, Modelling and Simulation*, (2016): 73-77. https://www.researchgate.net/publication/315364724_Chatbot_Using_a_Knowledge_in_Database_Human-to-Machine_Conversation_Modeling
14. Teller, Seth, Antone, Matthew, Correa, Andrew, Davis, Randall, Glass, Jim, Fletcher, Luke, Frazzoli, Emilio, Walter, Matthew R.. “A Voice-Commandable Robotic Forklift Working Alongside Humans in Minimally-Prepared Outdoor.” https://people.csail.mit.edu/teller/pubs/TellerEtAl_ICRA2010.pdf
15. Correa, Andrew, Davis, Randall, Glass, Jim, Fletcher, Luke, Teller, Seth, Walter, Matthew R.. “Multimodal Interaction with an Autonomous Forklift.” *IEEE*, (2010): 243-250. <https://people.csail.mit.edu/jrg/2010/Correa-HRI10.pdf>
16. Delden, Sebastian, Hess, Gregory, Rosario, Carlos, Umrysh, Michael. “Pick-and-place application development using voice and visual commands.” *Industrial Robot: An International Journal*, (2012): 592-600.
17. Pires, J. Norberto. “Robot-by-voice: experiments on commanding an industrial robot using the human voice.” *Industrial Robot: An International Journal*, (2005): 505-511. https://www.researchgate.net/publication/233608036_Robot-by-voice_Experiments_on_commanding_an_industrial_robot_using_the_human_voice

18. Bradshaw, John. "The Benefits of Voice-Directed Technology for Maintenance & Inspection Tasks | Manufacturing & Logistics IT Magazine." <https://www.logisticsit.com/articles/2017/06/15/the-benefits-of-voice-directed-technology-for-maintenance-and-inspection-tasks/>, 2017.
19. Pease, F. "The Future of Manufacturing: Voice-Activated Services for the Industry – Industry & Automation – Pictures of the Future – Innovation – Home – Siemens Global Website." <https://www.siemens.com/innovation/en/home/pictures-of-the-future/industry-and-automation/the-future-of-manufacturing-dialogue-manager.html>. 2017.
20. Sphinx, 2.1.2.NLTK Library <https://www.nltk.org/> © Copyright 2020, NLTK Project Mar 08, 2020.
21. Tensorflow.org. <https://www.tensorflow.org/> (2020)
22. Baby, Cyril Joe, Khan, Faizan Ayyub, N. Swathi, J. "Home Automation using IoT and a Chatbot using Natural Language Processing." International Conference on Innovations in Power and Advanced Computing Technologies, (2007): 1-6.
23. SpeechRecognition.PyPi, <https://pypi.org/project/SpeechRecognition/>, 2018. Stiefelhagen, R., Fugen, C., Giesemann, P., Holzapfel, H., Nickel, K. and Waibel, A..
24. Gnjatović, Milan, Borovac, Branislav, Delić, Vlado, Mišković, Dragiša, Nikolić, Milutin, Tasevski, Jovica. "Adaptive Multimodal Interaction with Industrial Robot." Adaptive Multimodal Interaction with Industrial Robot(2012): 329-333. https://www.researchgate.net/publication/251881852_Adaptive_Multimodal_Interaction_with_Industrial_Robot



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details