iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.46959

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Next Message Prediction in a Sequence using Machine Learning

Nikhil Anand Mahendrakar
Dept of Information Science and Engineering, BMS Institute of Technology

Abstract: ***Prediction of next message after given a sequence of messages has lot of applications. It has applications in NLP (natural language processing), Log Anomaly Detection, Automating Customer Service and many more. Automating Customer Service is one such application which shall be in great focus in this paper. For any organization, automating customer service can save a lot of revenue and time. It increases the response time for customers/users, which in turn increases customer satisfaction. It reduces the load on the business. In this paper, we use machine learning algorithm (Support vector machine + gridsearchcv) to automate the next debug message or step that has to be followed to solve a customer problem/ticket. This algorithm can be used to predict multiple next steps also, thereby automating the debug steps handled by the customer service team. It predicts the next debug message given a sequence at an accuracy of 98.08 percent.***
Keywords: Anomaly detection; log data analysis; sequence prediction

I. INTRODUCTION

Customer service automation is the process of automatically resolving user problems without human-to-human interaction. Customer Service team of any organisation deals with a lot of queries every day. Most of those requests are repetitive and usually solved. If automation of repetitive queries can be achieved, then the customer service team can only cater to more complex queries that requires human intervention. The customer service team can be small and can deal with problems that need specialized care. A small customer service team can be a great cost-saving mechanism for businesses.

Generally, the operations of a customer service team involves giving the customers/users debug messages/steps to solve the problem. This paper elaborates a method to develop a model which predicts the next step/debug message. The initial input for the model is the first debug message sequence and after that it can be used multiple times to generate the next debug messages/steps required to the solve the customer issue. To train the model, we have used event log of information technology incident management process.

II. RELATED WORKS

In this paper [1], the author has proposed two systems to predict for log analysis. The first system is probability based where the occurrence of the debug messages/steps is calculated and shown. The other system uses deep learning mechanism to predict the next log message. The deep learning algorithm used to build the model is LSTM (long short-term memory).

In this paper [2], the author has presented a approach to use unlabelled data to improve the prediction of next log message using recurrent networks. The author has followed two steps to improve the prediction. The first step is the standard prediction of the next message after a sequence using recurrent networks. In the next step, author has used the auto-encoder technique to give the input sequence as a vector and it produces the output sequence. This step is the pre-training phase of the model. This two step approach has improved generalization and prediction.

In this paper [3], the author has learned a Finite State Automation (FSA) the normal workflow of the system using the log sequences. There is a performance measurement model which classifies whether the execution of a particular log sequence is normal or a irregular sequence.

III. DATA

A. Structure of Data

Data is an event log of information technology incident management process [4]. It contains the various debug messages/step that has been followed to close the customer ticket. The total number of debug messages/steps that has to be followed to close a customer ticket varies from 1-78. Data is present in a CSV format. There are approximately 4.6 million rows and 5 columns. The 5 columns are Ticket Number, Status, time, group name and Owner. The status represents the most important information that shall be used for developing the model. The Figure 1 shows sample dataset.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

	A	B	C	D	E
1	TicketNum	Status	Time	GroupName	Owner
2	5078917	Open	7/1/2010 0:00	GRP01	RES01
3	5078917	acknowledged notification	7/1/2010 5:18	GRP01	RES11
4	5078917	Assignment	7/1/2010 5:21	GRP01	RES21
5	5078917	analysis/research/tech note	7/1/2010 5:21	GRP01	RES21
6	5078917	Assignment	7/1/2010 16:37	GRP01	RES31
7	5078917	pending customer	7/1/2010 16:37	GRP01	RES31
8	5078917	Assignment	7/6/2010 16:40	GRP01	RES31
9	5078917	reassigned-misrouted	7/6/2010 16:40	GRP01	RES31
10	5078917	reassigned-misrouted	7/6/2010 16:40	GRP01	RES31
11	5078917	acknowledged notification	7/6/2010 16:42	GRP01	RES31
12	5078917	Assignment	7/6/2010 16:54	GRP01	RES31
13	5078917	analysis/research/tech note	7/6/2010 16:54	GRP01	RES31
14	5078917	restored to service	7/6/2010 17:54	GRP01	RES31
15	5078917	Closed	7/6/2010 18:00	GRP01	RES31
16	5078921	Open	7/1/2010 0:01	GRP11	RES41
17	5078921	acknowledged notification	7/1/2010 0:02	GRP11	RES51
18	5078921	analysis/research/tech note	7/1/2010 0:02	GRP11	RES51
19	5078921	Closed	7/1/2010 0:25	GRP11	RES51
20	5078922	Open	7/1/2010 0:01	GRP11	RES41
21	5078922	acknowledged notification	7/1/2010 0:02	GRP11	RES51
22	5078922	analysis/research/tech note	7/1/2010 0:02	GRP11	RES51
23	5078922	Closed	7/1/2010 9:42	GRP11	RES51
24	5078971	Open	7/1/2010 1:09	GRP21	RES61
25	5078971	Assignment	7/1/2010 1:10	GRP21	RES11
26	5078971	analysis/research/tech note	7/1/2010 1:10	GRP21	RES11
27	5078971	alert stage 1	7/1/2010 1:30	GRP21	RES11

Figure 1

B. Data Preprocessing

The data is converted to fit the model. In this process, the data is first divided into sequences and is embedded in the dictionary. The figure 2 shows the code for achieving the abovementioned step. Once the dictionary has been populated with data, the next step is conversion of the string data into numerical form. For this, we get the unique debug messages/steps from the dataset and give numerical value. Figure 3 shows the unique debug messages/steps that has been used in solving customer ticket.

```
for row in csvreader:



    if row[1]=="Status":
        continue;

    listofkeys.append(row[1])
    if row[1]=="Open":
        i=i+1
        templist = []
    elif row[1] == "Closed":

        datasequence[i] = templist
        #print(datasequence)
        #print(templist)


    else:
        templist.append(row[1])
```

Figure 2

```
['', 'pending release', 'communication with customer', 'Open', 'pending repair', 'pending confirmation', 'Work in Progress', 'analysis/research/tech note', 'pe
nding vendor', 'Manually Acknowledged', 'Acknowledged', 'Pending Vendor', 'restarted notification', 'communication with provider', 'alert stage 1', 'Assignment
', 'Closed', 'automated', 'acknowledged notification', 'reassigned-misrouted', 'alert stage 2', 'Restored to Service', 'equipment return', 'Pending Customer',
'reassigned-addl work required', 'communication with vendor', 'waiting parts', 'restored to service', 'pending provider', 'pending customer', 'status request',
 'DEADLINE ALERT', 'alert stage 3']
```

Figure 3

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

IV. IMPLEMENTATION

A. *Scaling of Data*

Scaling of Data is an important process that has to be completed before we train the model. Scaling can be defined as a process to convert the data into a standardized form. Consider an example where there is involvement of two currencies: yuan and US dollar. Let's say the 1 US dollar = 7 yuan. When the operations are performed on these two currencies, it is important that they be converted to the same unit (in this case, yuan or US dollar) for effective mathematical operations. Figure 4 shows the code snippet for scaling of data.

```
st_x= StandardScaler()
x_train= st_x.fit_transform(X_train)
x_test= st_x.transform(X_test)
parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}
svc = svm.SVC()
clf = GridSearchCV(svc, parameters).fit(x_train,Y_train)
```

Figure 4

B. *Training the Model*

1.) *Assigning keys to the unique debug messages/steps*: As shown in figure 3, the unique debug messages/steps are assigned unique keys. These keys are the classes that we shall aim to predict. As shown, there are multiple keys/classes and hence it is a multiclass classifier problem.

2.) *Support Vector Machine (SVM):* It is a supervised machine learning algorithm which can be used for both binary and multiclass classifying problem. It is one of the most robust prediction technique which uses the VC theory (statistical learning framework) for classifying.

3.) *Grid Search CV*: For any model, the performance of the model depends on the hyperparameters chosen by the engineer. It is also important to note that, there is no way to precisely chose hyperparameters for a model beforehand. GridSearchCV is algorithm which is used to find the best hyperparameters for a algorithm. As mentioned above, the algorithm used is support vector machine and figure 5 shows the hyperparameters for support vector machine[5]. GridSearchCV finds the best hyperparameters by trying all the combinations and evaluates the model by cross-validation. Figure 6 shows the code snippet of fitting the data using gridsearchCV.

```
{ 'C': [0.1, 1, 10, 100, 1000],
 'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
 'kernel': ['rbf','linear','sigmoid'] }
```

Figure 5

```
svc = svm.SVC()
clf = GridSearchCV(svc, parameters).fit(x_train,Y_train)
```

Figure 6

V. RESULTS

The robustness of any model is tested using the f1-score, precision and recall score. Precision score signifies the number of positive class prediction that actually belong to the positive class. Recall signifies/quantifies the total number of correct positive class prediction. F1-measure provides a numerical number which takes precision and recall score both into consideration [6]. Confusion matrix displays all these information in a tabular structure as shown in the figure 7.

Accuracy score is one more important criteria which is considered to judge the efficiency of a machine learning model. The accuracy is defined as the ratio total number of correct prediction to total number of data points multiplied by 100. The accuracy of the model is 98.08 percent as shown in figure 7.

accuracy is 0.9808823529411764

	precision	recall	f1-score	support
2	1.00	0.91	0.95	43
3	1.00	0.97	0.99	34
5	1.00	1.00	1.00	14
6	1.00	1.00	1.00	10
8	1.00	0.92	0.96	51
9	1.00	1.00	1.00	1
13	1.00	1.00	1.00	2
15	1.00	1.00	1.00	1
19	1.00	1.00	1.00	1
25	1.00	1.00	1.00	1
28	0.98	0.98	0.98	219
30	0.97	1.00	0.99	303
accuracy			0.98	680
macro avg	1.00	0.98	0.99	680
weighted avg	0.98	0.98	0.98	680

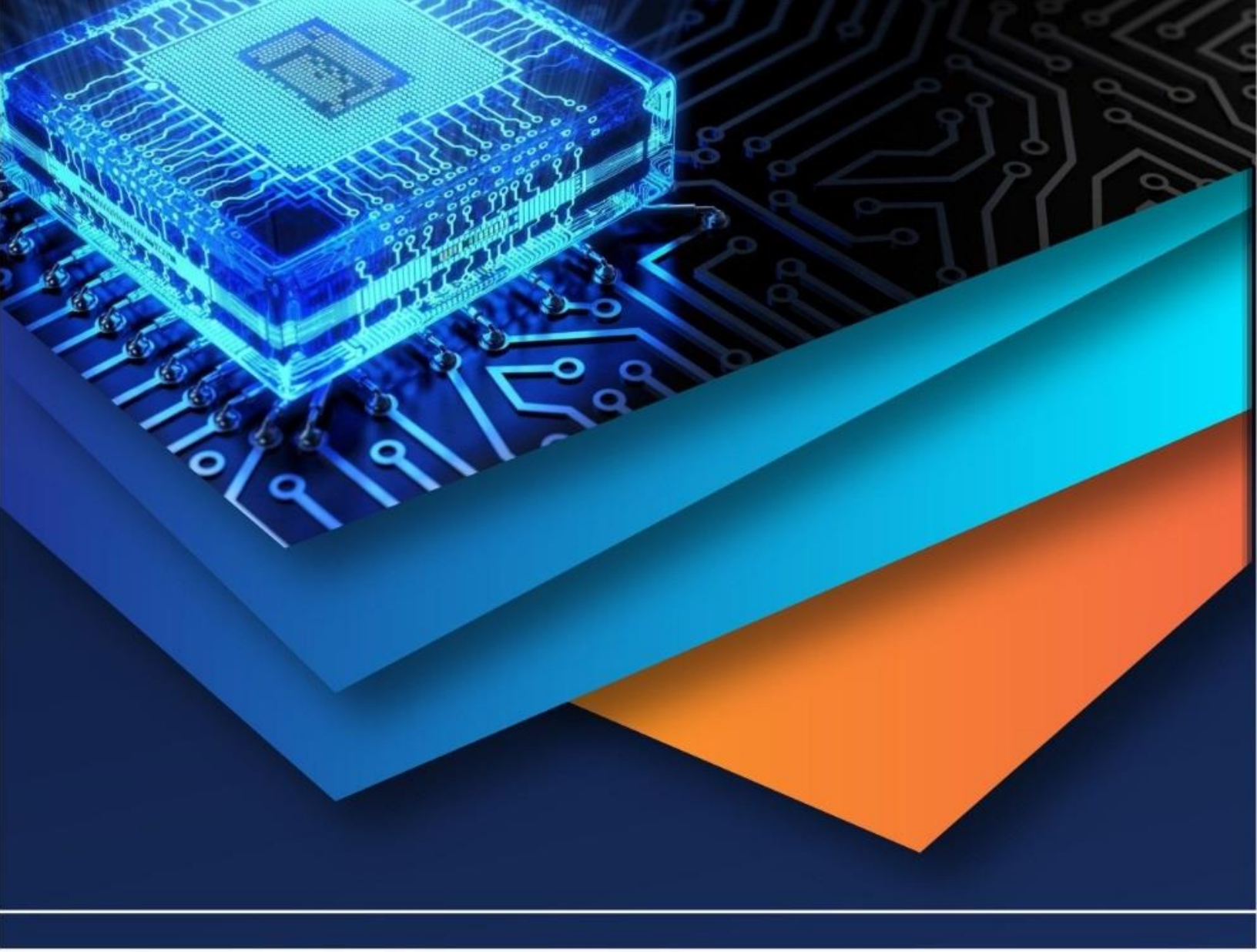
Figure 7

VI. CONCLUSIONS

Predicting the next debug messages/steps for any customer ticket has great advantages to business. In this paper, the model trained predicts the next debug message at very high accuracy of 98.08 percent. The total number of cases or issues with which the model has been trained is over 54000. Each case/issue has more than 55 debug messages/steps that the customer service team has given to the customers to close the ticket. The model has been trained comprehensively with many issues. This model can be run into a loop until it close is predicted and used as complete automating tool for generating the debug messages.Thus, it can be concluded that the above technique of using support vector machine with gridsearchcv can be used to automate the issues that are repetitive and that have already been closed.

REFERENCES

[1] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar,"DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 1285–1298. https://doi.org/10.1145/3133956.3134015.
[2] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning," Advances in neural information processing systems 28 (2015).
[3] Q. Fu, J. -G. Lou, Y. Wang and J. Li, "Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis," 2009 Ninth IEEE International Conference on Data Mining, 2009, pp. 149-158, doi: 10.1109/ICDM.2009.60.
[4] Event log of IT Incident management process. [Online]. Available: https://www.kaggle.com/datasets/asjad99/it-incident-management-process
[5] Hyperparameters for support vector machine (SVM). [Online]. Available: https://www.mygreatlearning.com/blog/gridsearchcv
[6] Definition of precision score, f1 socre and recall score. [Online]. Available: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY