Exploring Deep Neural Networks for Automated Ticket Classification

First and Corresponding Author:
Dr.PARAMESH S.P
Assistant Professor,
Department of CS & E,
School of Engineering
Central University of Karnataka
Gulbarga, Karnataka
Email: sp.paramesh@gmail.com

Second Author 2:
SHREEDHARA K.S
Professor,
Department of Studies in CS & E,
U.B.D.T College of Engineering
Davanagere, Karnataka

Abstract:
Manual categorization of service desk tickets may lead to dispatching of problem tickets to aninappropriateexpertgroup,delays the response time and interrupts the normal functioning of the business.Traditional machine learningapproachescan be applied to train an automated service desk ticket classifier by mining the historical tickets.Sparsity, non-linearity, overfitting and handcrafting of features are some of the issues concerning the traditional ticket classifiers.This research work proposes a deep neural network model based on Convolution Neural Network (CNN) for the automated classification of service desk tickets.A real-world service desk ticket data is used to corroborate the efficiency of the proposed ticket classifier model and compared the results with the traditionalclassifiers.The proposed CNN model outperforms the chosen alternatives in terms of overall model performance.Benefits of the proposed model includes assignment of tickets to the correct domain groups,speedy resolution, improved productivity, uninterrupted business and customer satisfaction.
Keywords: Service desk systems, Machine learning, Deep learning, Convolution Neural Network, Word Embeddings, Text classification, Feature maps.
DOI Number: 10.48047/NQ.2022.20.13.NQ88477 **Neuroquantology 2022; 20(13): 3922-3933**

1. Introduction

In the modern service delivery business, it is very much important to ensure that the required services are delivered to the end users of the organization on time for the smooth functioning of the business [1].Service desk or Helpdesk systems are the places where different stakeholders of the business can raise various kinds issue tickets and get the resolutions for their problems [2]. Service desk systems are also called as customer service desk, support desk or incident management systems.

The users of the organization may submit the problem tickets through various service desk platforms like email, chat, web-based user interface (UI) or through telephones. The designated service desk agent then analyses the submitted tickets, categorises the tickets manually and assign the ticket to the concerned domain teamon the basis of problem category for further resolution.The basic workflow in a classicalservice desk system is shown in Fig.1.

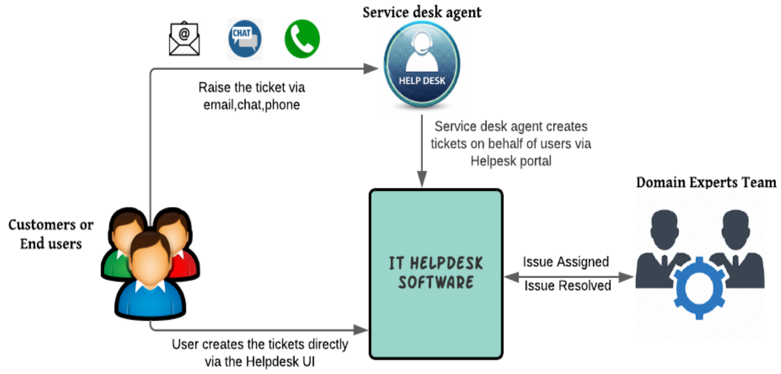

Fig.1 Basic workflowin a typical Service desk system

In some the web-based service desk systems, the end users create the issue tickets manually by selecting appropriate category using the web user interface. Depending upon the selected category, the tickets are forwarded to the concerned domain team for the resolution. The web UI of atypical service desk system is given in the Fig.2.

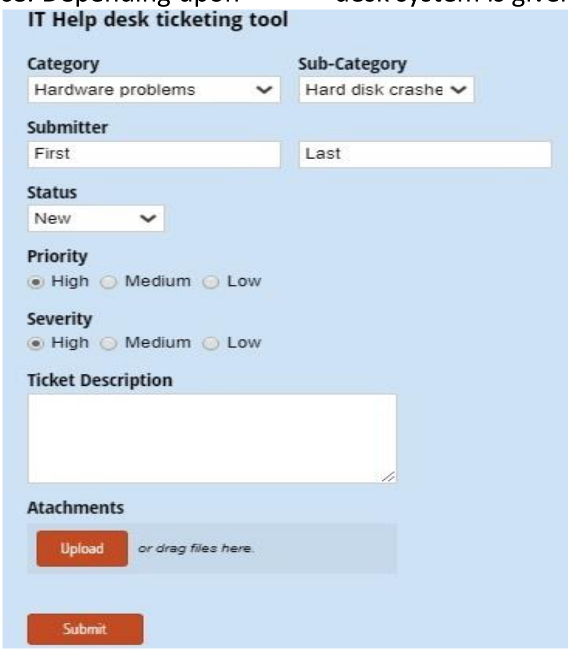

Fig.2 The Web user interface of Service desk system

Manual selection and categorization of problem tickets either by the service desk agent or by end users is time consuming process and it may result in wrong classification. Huge volumes of tickets, lack of domain knowledge, wrong perception about ticket category etc are some of the reasons for wrong categorization.Misclassification further results in assigning the tickets to wrong resolver group, interrupts the normal functioning of the business, unnecessary domain resource utilization, customer satisfaction deterioration and affects the business. Because of these limitations with the manual categorization oftickets, it is necessary to develop an intelligent model which auto categorises the supporttickets [3].

In the literature, combination of classical supervised machine learning (ML) and Natural Language Processing (NLP) techniques are explored to automate the process of ticket classification[3-9].Most of these approaches uses the historical unstructured natural language ticket descriptions and the corresponding labels for building the ticket classifier.The ticket classifier models based on traditional ML algorithm uses avector space model to representeach user's ticket descriptions which may result incomplicated model because of the increased dimensionality and sparsity of the data [12]. These models do not take into consideration the semantic similarity exists among various features in the dataset [4].Further,the handcrafted feature selection and extraction techniquesare used in these classicalmodelsto choosethe key features needed for the prediction [10,11].

To overcome these limitations of traditional machine learning ticket classifiers,a hierarchical deep neural network modelis proposed in this research work that uses word embeddings, CNNand a fully connected classification layercombinations to build the automated ticket classifier system.

Deep learning approaches uses the word embeddings to represent each ticket descriptions [16]. In this work, word embeddings are initialized randomly and learnt while training the neural network model.CNNsare effectively used in sentence and document classification problems due to their ability to extract automatically the most salient and highly representative n-grams from the input sentences. [13,14,15].Because of the dense representation of word embeddings and effective feature extraction function of CNN, it motivated us to use these combinations for building an effective automated ticket classifier.

The remainder of this paperisstructured as follows. Literature review in the field of automatedservice desk ticket classification and recent developments in the text document classification problem is discussed in section 2.Proposed methodologyforservice desk ticket classification using CNN based deep neural network is described in section 3. Section 4 reportthe experiments conducted and results obtainedusing the training and test dataset. In the last section we conclude the present research work with some future research directions.

2. Prior research works

Some of the priorresearch effortsin the development of automated service desk ticket classifiers systems is given below.

Al-Hawari et al. [3] discusses the development of IT helpdesk ticket classifierusingthe traditionalML methods like J-48, Rule based, Support Vector Machines (SVM) and Naive bayes. The performances of the models isvalidated against the 331 helpdesk test tickets and it is found that the SVM algorithm outperformed its counterparts.

Jian Xu et al. [4] proposes a support ticket classification model to automatically identify the category of an input ticket. The proposed framework consists of modules for data preparation, partitioning of tickets and signature construction followed bysignature-basedticket classification. Data preparation phase removes the noises in the historical ticket data and gets the proper representation for each ticket.The partition and signature construction algorithm generatesvarious ticket groups and correspondingsignatures.A classification algorithmthen determines the label of unseen databy finding the most similar group signature.

Paramesh et al. [5] developedan automated service desk ticket classifier using classicalapproaches like Logistic regression, SVM, Naive bayes and K-Nearest Neighbour (KNN). The model performance is evaluated using a real-world IT infrastructure data containing ticket descriptions and the ticket label. SVM outperformed other chosen alternatives with respect to overall performance. Various data related challenges found in the historical dataset and the techniques used to handle these challenges are discussed in this work.

Mandal A. et al [6] proposes an email ticket dispatcher system usinga combination of ticket classifier ensemble and a rule engine. The email ticket classifier models are generated using the historical email data containing the email subject, descriptions and corresponding resolver group. The rules engine is used to capture the domain specific features from the training data.

Automated ticket classification is an extension of text document classification problem and hence literature review of the text classification approaches is done and some of these works are detailed below.

Kowsari et al. [12] provided a good review on the various techniques involved in the text classification process.Methods for text data pre-processing, dimensionality reduction, various ML techniques used to build text classifier and model evaluation metrics are discussed in detail.

Collobert and Weston [13] proposed a unified architecture using a single layer Convolution Neural Network for modelling the sentence. Given an input sentence, this model implements the multiple NLPproblems like Chunking, NER,PoStagging,semantic roles labelling etc.

Kim [14] explored a Single layer CNN architecture built on top of pre-defined word vectors (like Word2vec or glove) for sentence classification problemslike sentiment analysis and question type classification. Many experiments were conducted with this architecture and the results shows that with little hyperparameter tuning and a predefined word embedding, the model can achieve excellentperformance on various sentence classification tasks.This model however, had some limitations withCNN's inability to model the long-term dependencies.

The issue of long-term dependency is partially addressed by Kalchbrenner et al. [15], where they proposed a dynamic CNN for modelling of the sentences. The model uses the k-Max pooling strategy to select the k-most salient features required for the modelling.

3. Proposed Research Methodology

The process of service desk ticket classification is an example of textdocument classification with the ticket descriptions and its associated category representing the text document and document label respectively.The proposed solution diagram for the automated service desk ticket classification using a CNN deep learning model isdepictedin Fig.3.

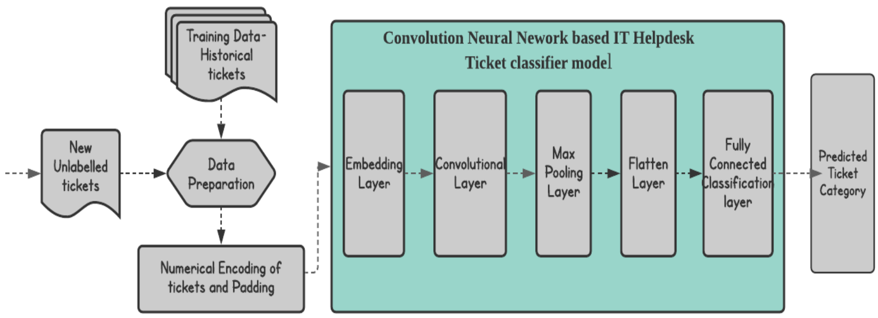

Fig.3 Proposedsolution diagram of CNN based Service desk ticket classification

The key components involved in the ticket classification using CNN are explained as follows.

3.1 Training data

In this work, a real-worldIT infrastructure ticket data is used for building theproposed ticket classifier.Hardware issues, LAN, OS, printer, email issues etc. are some of the categories associated with infrastructure data. The historical tickets containing the unstructured ticket descriptions and the associated category are used to build the ticket classifier model.

3.2 Pre-processing of Service desk tickets

Data pre-processing is one of the most crucialphase in any typical data mining process to build an efficient ML model.It involves removing any undesired and other noisy data from the dataset to get the most relevant attribute set.Many data related issues and challenges were found in the initial raw data chosen for this work. The various issues found and the techniques used handle these flaws are detailed below.

3.2.1 Handling of large number of classes

The dataset considered for our research work had large number of ticket categories.Some of these classes were correlated with each other and syntactically similar. Having many numbers of classes with similar intent may lead to assigning the service tickets to improper resolver group and unnecessary resource utilization.So similar classes were merged manually to minimize the number of categories of the dataset.Clustering methodsalso leveraged to identify the similar ticket instancesand the clusters containing these instances are manually labelled with help of domain knowledge.

3.2.2 Handling the class imbalance problems

The chosen dataset had many imbalanced classes and this may cause the training model to become biased towards the ticket categorycontaining morenumber of tuples. The below mentioned techniques were used to overcome class imbalance problems.

- Classes with less number of tickets (for ex: having 1 or 2 instances per class) can be ignored and only thosecategories having a minimum threshold value are considered for the training purpose.
- Data sampling methods like under sampling and over sampling.

3.2.3 Handling the noisy ticket data

Huge amount of unwantedand other kinds of messy data were present in the initialraw dataset.To remove such undesired data, each ticket descriptions are converted into lower case and further tokenized into words, numbers, punctuations, email ids etc using the tokenization process. Initially, all the commonly used English words or stop words are removed. Entities like email ids, date, time, special characters etc are removed by developing appropriate pattern recognizers. Some of the functional words like verbs, nouns, pronouns, conjunctions, prepositions and quantifiers present in the ticket descriptions are removed by tagging each word with a Parts of Speech (PoS) tagging.Finally, stemming operation is performed, which reduces each feature to its base form.

The various data related issues and challengesfound in our chosen dataset may appear in other helpdesk ticket datasets also irrespective of the domain area. The different techniques discussedin this work to handle such issuescan be used in general for other datasets also along with the use of other appropriate methods depending upon the dataset.

3.3 Numerical encoding and padding of ticket descriptions

CNN model accepts only fixed length numerical input data.Each pre-processed ticket descriptionsaretokenized into words followed by integer encoding of the words based on the word index in the entire corpus of training data.The encoded ticket descriptionsare then padded using padding mechanism to ensure every input is equal in size.In this research work, we fixed each ticket description to contain a maximum of n =250 words and this length'n'is determined by finding the ticket instance containing the maximum number of features.

3.4 CNN based deep learning model for service desk ticket classification

The proposed CNN model for the Service desk ticket classification comprises of aninput embedding layer, single dimensionconvolution layer, pooling and a fully connected classification layer. Each of these layers are described below.

3.4.1Embedding Layer

Numerically encoded and padded input sentence (i.e., a ticket description in our case)is fed as an input to the embedding layer.Let a ticket instance 'S'consisting of 'n' words be the input to the model.Let $x_i \in \mathbb{R}^d$be the d-dimentional real valuedfeaturevector of the ith tokenof the sentence and is represented as:

$x_i = [r_1, r_2, \ldots\ldots r_d]$(1)

We used a word embedding vector dimension of d =100 and these vectors are initialized with random weights and are learned during the neural network model training.The sentence with 'n' words is represented using anembedding matrix or input sentence representation matrix $S \in \mathbb{R}^{n \times d}$. The embedding matrix $S$is formed by concatenating the d-dimensional real valued vector of x_i for every$i$th token in the sentence S and is represented as:

$S = x_{1:n} = [x_1 \oplus x_2 \oplus x_3 \ldots\ldots\ldots \oplus x_n]$(2)

Here, $\oplus$ is the concatenation symbol. So, the dimension of our input embedding matrix is $n \times d$ =250x100.

3.4.2 Convolution Layer

Convolution layer is used as feature extraction layer which extracts the most salient features from the input ticket descriptions represented using the word embeddings. Input embedding matrix of size $n \times d$corresponding to each ticket description $S$is given as an input to the convolution layer.Convolution operation is performed on the embedding matrix using a linear filterof some fixed height 'h'. A convolution involves a filter parameterized by a weight matrix 'm' belongs to $\mathbb{R}^{h\times d}$and is applied over'h'input words to generate a new feature. An element$c_i$created using a window of features$w_{i:i+h-1}$ is given by

$$c_i = f(w_{i:i+h-1} \cdot m + b) \quad (3)$$

Here '$\cdot$' is element wise product operator, $b \in \mathbb{R}$ is the bias value and $f$denotes a non-linear activation functionlike Rectified Linear Unit (ReLU). The filter vector 'm'convolves over all possible input features in the sentence to produce a new single dimension feature map

$$\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-h+1}] \quad (4)$$

with $\boldsymbol{c} \in \mathbb{R}^{n-h+1}$.

Kernel size represents the number of n-grams to consider during convolution i.e., filter size. One may use number of filters of different kernel sizes where eachfilter extracts a particular pattern of n-gram.

In this work, we used multi-channel CNN model with different kernel sizes$h$ =3,4, and5which convolves respectively over3,4 and 5 words of the input at once to generatedistinct feature mapshaving size (*n*-*h*+1)×1. The number of convolution filters for each kernel size is set to 32. The best kernel sizeand the optimal number of features maps to use for each kernel size is chosen after fine tuning the hyperparameters of CNN model during the training process.

3.4.3 Pooling layer

The feature maps generated by the convolution layer are fed to the pooling layer to extract the most relevant features from each map. One-dimensionmax pooling functionis applied on every map which extracts thefeature with the highest value$\hat{c} = \max\{\boldsymbol{c}\}$from each feature map.Since we used '32' different feature maps for each kernel sizes h =3,4 and 5, a total of '96' most important features are extracted during pooling operation. The advantages of max poolingare it extracts most relevant features across the whole ticket instance, decreases the computational overhead of the network and controls the overfitting.

3.4.4 Fully connected Classification layer

The salient features generated using the 1-max pooling are concatenated into a single dimensional feature vector which is fed directly intooutput classification layer for prediction.The classification layer gives the output in the form of a vector the size of which is equal to the number of target classes.In the present research work, because of the multiple classes in the dataset, we used softmax activation functions on each unit of the output layer.At this layer, one may apply 'dropout' as a means of regularization [17]. The softmax function gives the probability distribution of each input ticket instance belonging to different classes of the training data and is given as follows.

3927

$$P(c = i|S, x, z) = softmax(x^T w_i + z_i) = \frac{e^{x^T w_i + z_i}}{\sum_{k=1}^{K} e^{x^T w_i + z_i}} \quad (5)$$

Where z_k and w_k are the bias and weight vector of the kth labels.

The error between the actual and predicted output is found using a suitable loss function.Theintent of the training is to optimize the loss function by updating the network parameters like filter weights, bias value of activation function and weight vectors of the softmax function using a suitable optimizer.We used *categorical_crossentropy* asthe loss function and is given by

$$L(\hat{y}, y) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j \log(\hat{y}_i^j) \quad (6)$$

Here, y_i^j and $\hat{y}_i^j$ represents the actual and predicted output. $N$is the sample size and $C$is the number of classes.In this work, a stochastic gradient optimizerand backpropagation learningis used for updating the network parameters [18].

As depicted in the Fig.4, the input embedding matrix corresponding to each ticket description is of $n \times d$= 250x100 dimension. We used three different filter sizes h =3,4 and 5 and 32 filters of each kernel size. Filters convolves over the input matrix and generates the feature maps of size $n - h + 1$. A total of 96 (i.e.,32x3) feature maps are generated in this case. 1-max pooling operation is done on each generated feature maps to extract the maximum value from each map. A univariate vector is generated from each feature maps

NeuroQuantology| October 2022 | Volume 20 | Issue 13 | Page 3922-3933| doi: 10.48074/nq.2022.20.13.NQ88477

and a total of 96 features are then concatenated to form a single dimension feature vector. Finally, thesoftmax classification layer uses this feature vector to classify the service desk tickets.

3.5 Proposed model performance evaluation

The accuracy is one of the primary metrics to summarize the classifier performance and is defined as the fraction of correct predictions over the total number of predictions made by the ticket classifier. Metrics such as precision, recall and f-score are also used along with accuracy to analyse the effectiveness of our model.

4.Experiments and Results

The current research work has been doneusing a computer system equipped with 10th generation Intel core i7processor, 8GB SD RAM, NVIDIA Graphics cardand Windows 10 with 64-bit OS.Python's Keras 2.3.1with TensorFlow as backend, Scikit-learn, Pandasand Matplotliblibraries are usedfor implementation.

4.1 Dataset

A real timeIT infrastructure service desk ticket data containing multiple number of classes is used for our experimental purposes. The dataset is collected over period of one month from a reputed IT organization's infrastructure service desk.Issues related to hardware, operating system, network, printer, software etc are some of the categories associated with the IT infrastructure dataset.Some statistics about the chosendataset are given in the Table.1. The various challenges found in the initial dataset and the techniques used to handle these challenges are discussed in detail in section 3.2.

Table.1. Statistics about the IT Infrastructure dataset

Total no of ticket instances in the initial dataset	≈13,000
No of actual target classesin the initialdataset	25
Total no of ticket instances generated after merging the similar classes and handling the class imbalance issue in the dataset.	≈11,000
No of target classes generated after merging the similar classes in the raw dataset	18
No of unique features in the dataset before pre-processing of data	7523
No of features in the dataset after removing the Englishstop words	7400
Vocabulary size after removingvarious unwanted features from the dataset.	3572
Training and test dataset size ratio	80:20

4.2 Numerical encoding and padding of ticket instances

As discussed in the section 3.3 of our research work, each pre-processed ticket descriptions are encoded as integers and are further subjected to padding to ensure each ticket instances are of fixedlengthi.e., n=250 in our case. The sample output screenshotafter encoding and pre-padding of each ticket instances is depicted in Fig.4.

```
[[   0    0    0 ...   13  630  183]
 [   0    0    0 ...    9    1  764]
 [   0    0    0 ...    0   30   37]
 ...
 [   0    0    0 ...    0   62   37]
 [   0    0    0 ...   89   67   10]
 [   0    0    0 ... 1356   63 1959]]
```

Fig.4 Encoded and Padded ticket instances

Each encoded and padded ticket instance is then converted into an embedding matrix wherein each feature is mapped to real valued vector of dimension d=100 so that the size of the input sentence matrix is $n \times d$= 250x100.The input embedding matrix corresponding each input ticket is then fed to the CNN based classifier model as an input.

4.3 Proposed CNN model hyper parameter settings

Several experiments were conducted to choose the best values of the filter sizes,

eISSN1303-5150

optimal feature maps, epochsand other hyper parameters. Different classifier aredevelopedusing the training data and with different filter sizeswhile keeping the feature map size=32.Mean accuracy performances of the training models obtained using different filter sizes is shown in Table.2.

Table.2 Mean Accuracy performance of the training model using different sized filters

Filter Sizes	Mean Accuracy
1	84.6
3	85.8
1,2	87.9
2,3	88.1
3,4	88.5
1,2,3	88.2
2,3,4	88.4
3,4,5	**91.0**

From the Table.2, it indicates that the best filter size combination is (3,4,5) with an average training accuracy of 91%. The hyperparameters of the best model used in this research workare givenin Table.3.

Table.3 Best hyperparameters for the proposedticket classifier model

Hyperparameter	Choice
Maximum length of anyinput ticket instance	250
Word Embedding size for each feature	100
Filter/Kernel size	(3,4,5)
Number of filters	32
Activation function used in CNN layer	ReLU
Pooling stride	1
Pooling strategy	1-Maxpooling
Learning rate	0.001
Loss function	*Categorical_Crossentropy*
Optimization algorithm	Stochastic Gradient Descent (SGD)
Epochs	30
Batch size	64
Activation function used in the output layer	Softmax

4.4 CNN Model building and Evaluation

The proposed ticket classifier model is trained using 80% of the original dataset and using the abovebest hyperparameters.After training the model with the best CNN hyperparameters, the loss and accuracy function plots between the training and validation set is shown in Fig.5 and Fig.6 respectively.

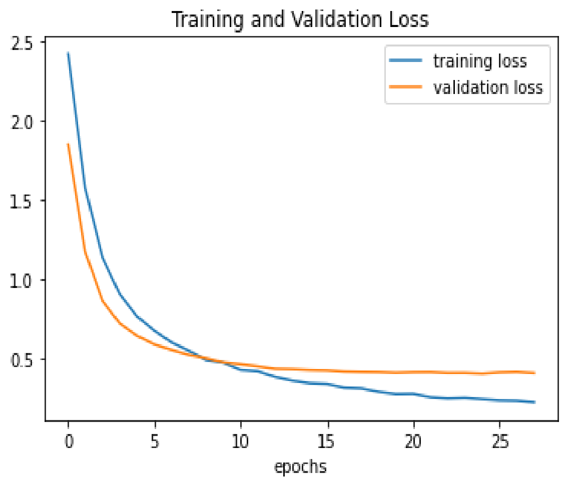

Fig.5. Training Vs Validation Loss

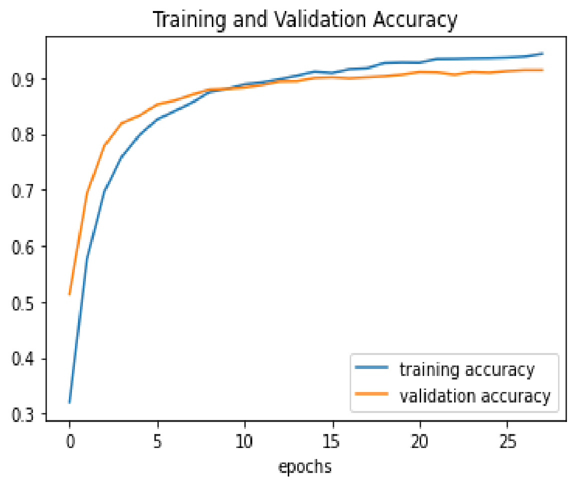

Fig.6. Training Vs Validation accuracy

The above two plots illustrates that ourmodel performed well with a training accuracy of 93% and quite low training lossof 0.15.Validation accuracy became stagnantafter 91% and doesn't increased much except at certain epochs. However, it looks like the model exhibits a little bit of overfitting with a validation accuracy of 91% and validation loss of 0.38.A dropout layer is added to the network to reduce this little overfitting problem.

Theeffectiveness of the proposed model effectiveness is also compared with traditional classifiers like SVM, Logistic Regression (LR), Multinomial Naive Bayes (MNB) and KNNusing various classifier metrics over the test set and the obtained resultsare shown in Table.4 and Fig.7.

Table.4: Performance comparison of the classical ML models and the proposed CNN model

Model	Accuracy	Precision	Recall	F-score
LR	81.87	81.64	80.87	80.53
KNN	69.35	73.75	68.35	68.53
MNB	69.40	73.53	68.40	65.40
SVM	87.43	88.46	88.43	88.36
CNN	**90.80**	**91.18**	**88.78**	**89.52**

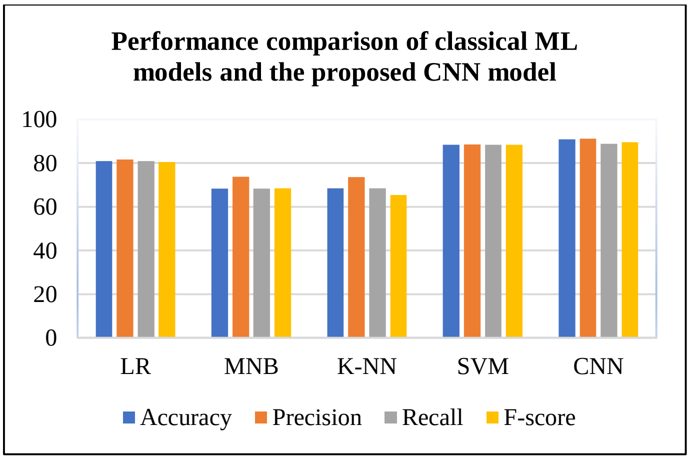

Fig.7. Performance comparison of ML models and the proposed CNN model

From the above results, it is found that proposed CNN based ticket classifier model with 90.80% accuracy outperforms the classical machine learning models in classifying the service desk ticketsfor the chosen dataset. A holistic view of the proposed model performance against test set is depicted in Fig.8 and Fig.9using classification report and the confusion matrix respectively.

The classification report of CNN based ticket classifier model on test data

	precision	recall	f1-score	support
BT MeetMe	0.31	0.97	0.47	38
Bitlocker issue	0.90	0.78	0.84	23
Browser	0.94	0.72	0.81	141
Citrix	0.91	0.83	0.87	107
Distribution Group	0.89	0.91	0.90	56
Hard Disk Issue	0.84	0.81	0.83	53
Hardware Problem	0.90	0.88	0.89	537
Network issues	0.90	0.97	0.94	39
OS Problem	1.00	0.83	0.90	23
Outlook issue	0.90	0.91	0.90	296
Printer Install/Issues	0.95	0.80	0.87	46
Project Specific Software	0.96	0.93	0.94	257
SCCM	0.92	0.93	0.92	86
Skype for Business	0.95	0.92	0.94	207
Unlicensed Software	1.00	1.00	1.00	37
Unlock/Password Reset	0.97	0.97	0.97	76
VPN issue	0.94	0.84	0.89	90
Virus Problem	0.92	0.95	0.93	37
accuracy			0.89	2149
macro avg	0.89	0.89	0.88	2149
weighted avg	0.91	0.89	0.90	2149

Fig.8 Classification report of CNN ticket classifier model on the test data

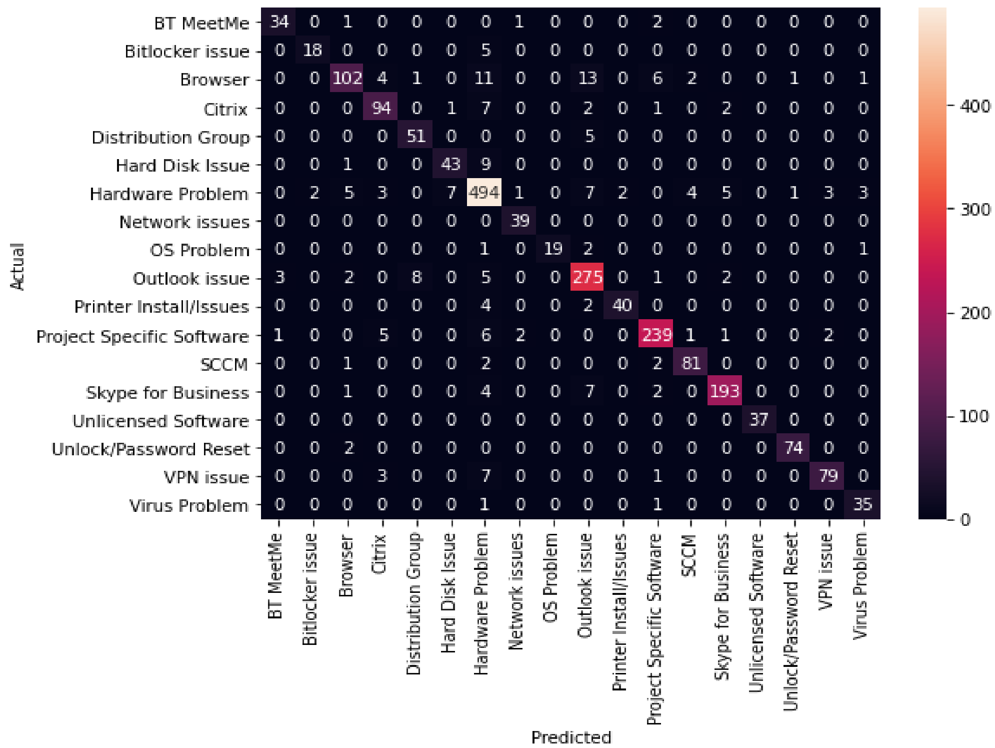

Fig.9 Confusion matrix of CNN ticket classifier model on the test data

5.Conclusion and Future work

To overcome the limitations of manual classification of service desk tickets, an automated service desk ticket classifier models can be developed by using machine learning techniques.A CNN based hierarchical deepneural network model is proposed in this research work for the autocategorization of service desk tickets. The proposed model isvalidated using a real-world IT infrastructure service desktickets and itachieved a training and validation accuracy of 93% and 91% respectively. The CNN model performance is also compared with the traditional classifiersand the results indicates CNN based classifier outperformed all other chosen alternatives with an accuracy of 90.8% in comparison with SVM (87.43%), MNB (69.4%), KNN (69.35%) and LR (81.87%). Effective resource utilization, quicker response time, improved end user experience, business growth etc. are some of the benefits of the proposed automated ticket classifier model. Though hierarchical CNN modelextracts the most informative features needed for classification and performed well in comparison to classical approaches, it doesn't effectively consider the long-range contextual dependencies exists in the unstructured ticket descriptions. Ourfuture work would focus on using the sequential deep neural network architecturesto capture the inherent sequential structure and to learn the long-term dependencies during the modelling of the automated service desk ticket classifier.

References

1. P. Kubiak, S. Rass, "An Overview of Data-Driven Techniques for IT-Service-Management," IEEE Access, 6, pp. 63664–63688, 2018.
2. M. Jantti, A. Cater-Steel, A. Shrestha, and N. Campus, "Towards an Improved IT Service Desk System and Processes: A Case Study," pp. 203-215, 2012.
3. Al-Hawari, F., Barham, H.: A machine learning based help desk system for IT service management. Journal of King Saud University - Computer and Information Sciences, 2019.
4. J. Xu, H. Zhang, W. Zhou, R. He, and T. Li, "Signature based trouble ticket classification" Future Gener. Comput. Syst., vol. 78, no. 1, pp. 41–58, 2018.
5. Paramesh S.P., Shreedhara K.S. "Automated IT Service Desk Systems Using Machine Learning Techniques". Proceedings of First International conference on Data Analytics and Learning, Lecture Notes in Networks and Systems, vol 43, pp.331-346, Springer, Singapore, 2019.
6. Mandal A., Malhotra N., Agarwal S., Ray A., Sridhara G. "Cognitive System to Achieve Human-Level Accuracy in Automated Assignment of Helpdesk Email Tickets", In: Pahl C., Vukovic M., Yin J., Yu Q. (eds) Service-Oriented Computing. ICSOC 2018. Lecture Notes in Computer Science, vol 11236. Springer, Cham, 2018.
7. Pandey, N., Sanyal, D.K., Hudait, A. et al. Automated classification of software issuereports using machine learning techniques: an empirical study. Innovations Syst SoftwEng 13, 279–297, 2017.
8. S. Roy, D. P. Muni, J.-J. Y. T. Yan, N. Budhiraja and F. Ceiler, "Clustering and labeling it maintenance tickets", International Conference on Service-Oriented Computing, pp. 829-845, 2016.
9. Cristina Kadar, Dorothea Wiesmann, Jose Iria, Dirk Husemann, and Mario Lucic. Automatic classification of change requests for improved it service quality. In Proceedings of the 2011 Annual SRII Global Conference, SRII'11, pages 430--439, Washington, DC, USA, IEEE Computer Society, 2011.
10. M. Ikonomakis, S. Kotsiantis and V. Tampakas, "Text Classification using Machine Learning Techniques", WSEAS Transactions on Computers, Vol. 4, No. 8, pp. 966-974, 2005.
11. Marcin MichałMirończuk, Jarosław Protasiewicz, A recent overview of the state-of-the-art elements of text classification, Expert Systems with Applications, Volume 106, Pages 36-54, ISSN 0957-4174, 2018.
12. Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, Brown, Text Classification Algorithms: A Survey, Information. 10 (2019) 150.
13. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, vol. 12, no. Aug, pp. 2493–2537, 2011.
14. Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.
15. N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, June 2014.

16. T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]," in IEEE Computational Intelligence Magazine, vol. 13, no. 3, pp. 55-75, 2018.
17. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” (2014)
18. D. E. Rumelhart, G. E. Hinton, and R. J. Williams,“Learning internal representations by error propagation,” DTIC Document, Tech. Rep., 1985.