Journal of
Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.776695

LEARNING LONG-TERM DEPENDENCIES FOR PREDICTION OF IT INCIDENT CATEGORY USING LSTM RECURRENT NEURAL NETWORKS

Dr. Paramesh S.P
Assistant Professor, Dept of CS & E, School of Engineering, Central University of Karnataka, Gulbarga, Karnataka, email: sp.paramesh@gmail.com

Abstract: Classification and routing of IT incidents to the proper domain expert team is a key issue in the IT service industry. Existing IT service management systems involves manual classification and routing of incidents. Manual classification of incidents may involve misclassification and hence results in assigning of the incidents to a wrong resolver group. Combination of natural language processing and supervised machine learning techniques can be leveraged to develop an automated Incident categorization system using a labelled training data. This research work proposes a methodology to develop an automated incident classifier based on deep learning using LSTM- RNN models by mining the natural language incident description entered by the user. LSTM deep learning architectures are effective in memorizing important information and efficient in learning the long-term contextual dependencies exists in the incident description. Word embedding representations are used in this research work to numerically encode the incident descriptions. The efficacy of the proposed incident classifier model is empirically validated using a real IT infrastructure incident data and compared the results with Support Vector machines, Logistic Regression, Naive Bayes and K-Nearest Neighbour. The study showed that LSTM-RNN model with optimal hyperparameters reported the best performance results based on various classifier performance measures. Proper assignment of incidents to the respective domain team, speedy resolution, improved productivity, increased customer satisfaction and uninterrupted business are some of the advantages of the proposed automated incident classifier model.
Keywords: *IT incident management, Machine Learning, Natural Language Processing, Long Short-Term Memory, Recurrent Neural Networks, Word Embeddings.*

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]. Information Technology Service Management (ITSM) is a framework for managing Information Technology (IT) operations and services in conformity with the needs of the business [2]. The ITSM framework aims to provide a high service quality to the end users of the organization and results in effective utilization of domain resources, customer satisfaction and increasing growth in business.

IT Incident management systems (helpdesk or service desk systems) are the important tools based on ITSM frameworks through which the end users can raise the issues related to their organizational services and get the resolution for the same [3]. In a service organization, IT incidents are created frequently and these incidents should be quickly resolved by the domain expert team for the proper functioning of the organization business. IT incident management is a component of ITSM process mainly concerned with managing the IT incidents from submission till the closure [4]. Submission of incidents, analysis and classification of these incidents by the service desk agent, assignment of tickets to domain group based on the incident category, providing proper resolution within the specified time period are some the important phases of an IT incident management. Incident management environments can be of web based, chat based, email or call-based systems [5].

Regardless of the type of the incident management systems, most of the service delivery systems involve manual classification of incidents either by service desk agent or by the end users. A high frequency of incidents in a service environment can generate an overwhelming number of incidents and it becomes difficult for the agent to handle, manually categorize the incidents and delegate it to the proper resolver group. On the other hand, direct users of the web-based systems may select wrong category due to lack of domain knowledge. Manual categorization of incidents is a time-consuming process and may involve wrong selection of incident category which in turn leads to dispatching the incident to the wrong resolver group. Misclassification further leads to incident reassignment, unnecessary domain resource utilization, resolution delay and customer satisfaction deterioration [6].

To address all these limitations of the existing traditional incident management systems and to be competitive, service organizations require an efficient service and support delivery systems. Machine learning (ML) and Natural Language Processing (NLP) techniques could be leveraged in the service management systems to automatize the process of incident classification and assignment in ITSM systems. Supervised ML algorithms and NLP techniques aid in building such an automated incident classifier [7-10]. These automated systems automatically classify the incidents by processing the incident natural language text description logged by end user. Historical incident dataset containing labelled incidents with its associated incident description play a useful role in building such a classifier system. Unsupervised machine learning techniques like clustering can also be leveraged to build the automated incident classifier when the training dataset contains unlabelled incident descriptions [11].

Sparse representation, overfitting, handcrafting of features, inability to consider the semantic similarity and contextual dependencies exist among the features etc are some of the issues with classical models [12,13]. So, to address the issues with the classical supervised ML models, deep learning-based models which are efficient in automatic extraction of salient features from the incident description data and are able learn long term contextual dependencies among the features of the data can be explored for building the efficient incident classifier models. The research work in [14] by the current author uses a deep neural network model based on Convolutional Neural Network (CNN) to model the automated IT service desk ticket classifier. CNN focuses on extracting the most salient features but the word ordering is restricted locally and long-term dependencies are ignored [15].

In this research work, a deep neural network model based on Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) is proposed to model the automated IT incident classifier. The proposed classifier model deals with the long-term dependency problem effectively and have not been explored in the area of automated ITSM systems. The proposed classifier auto categorises the IT incidents into one of the predefined service categories by mining the unstructured natural language description of the incident. The current research work uses the word embedding representation method to represent each incident descriptions since the deep neural network models do not support the use of sparse representation of the data. The proposed LSTM model used for incident classification is based on the architecture adopted by Hochreiter & Schmidhuber [16].

An IT infrastructure incident dataset containing labelled instances is used for experimental purposes in this research work. Hardware, local area network, wifi issues, software installation, email issues etc are some of the problems that can be raised under the IT infrastructure category. The structure of the typical IT infrastructure incident data raised by employees of the organization is given in Table 1.

Table 1. Sample IT Incident data

Incident ID	**Category**	**Priority**	**Severity**	**Submitter**	**Incident description**	**Status**
201	LAN	Medium	High	ABC	Unable to login into LAN	Closed
202	Hardware	Medium	Medium	PQR	Hard disk crashed	In progress
203	OS related	High	High	XYZ	Need to install OS	Open

The dataset considered for this research work contains fields like category, submitter, severity, priority, incident description fields etc but user's unstructured natural language incident descriptions and its corresponding categories are mainly used to build the proposed automated incident classifier model. The initial raw dataset used in this work had many irrelevant and noisy data. Appropriate pre-processing techniques are used to clean all such unwanted data. The major research contributions of this paper are –

- Pre-processing of the chosen IT service incident dataset.
- Numerical encoding of the incident descriptions using word embedding representation.
- LSTM model training, hyperparameters tuning and performance evaluation of the model for autoclassification of IT incidents.
- Comparison of the proposed LSTM model with traditional ML algorithms using various classifier evaluation measures.

Quick resolution time, effective resource utilization, improved productivity, uninterrupted service delivery and growth in organization business etc are some of the benefits of the proposed system.

2. RELATED WORKS

Although many prior research works were carried out in the field of text mining applications using ML and NLP techniques, a fewer number of research works have been done in the field of automation of IT Incident management systems. Some of the previously performed research efforts related to the automation of incident classification in IT service management system are given below.

Mucahit Altintas et al. [5] proposed a helpdesk ticket classifier using different supervised ML algorithms. The ticket classifier is developed by mining the natural language text descriptions of the tickets. Unwanted entities of ticket description like stop words, special characters etc. are removed during pre-processing phase. The tickets are represented using feature vector based on Binary, Term Frequency (TF) and TF-IDF term weighting schemes. To classify the tickets, authors experimented with 4 algorithms namely SVM, Naive bayes, KNN and Decision tree. SVM achieves the higher performance, with 86% accuracy compared to other chosen alternatives. Results indicates that the model performance depends on the chosen dataset, term encoding metric and ML algorithms used to develop the model.

Feras Al-Hawari et al. [6] developed a helpdesk ticket classifier model to assign a helpdesk ticket to the proper domain team to minimize the ticket resolution time and to save the operational resources. IT helpdesk ticket classifier model is developed using J48, Decision Table, Naive Bayes and SVM algorithms. The model performance is validated against the 331 helpdesk test tickets. Linear SVM algorithm with 81.4% accuracy outperformed all other classifier models.

Harun et al. [7] developed a framework for automated question classification in the helpdesk ticketing system using ML techniques. Supervised ML methods like SVM and Naive Bayes (NB) are used to build the classifier. The proposed framework results in associating the service ticket to the correct resolver group, faster resolution, efficient human resource utilization and increased user satisfaction.

A. Revina et al. [8] gives a comparative analysis of different ticket representation methods while developing an IT helpdesk ticket classification model. Real time ticket datasets containing the ticket description and label field sourced from an ITIL Change management (CHM) department of a large organization is used for the research work. Vector space model based on tf-idf term weighting scheme and domain related features are used to represent the tickets. The ticket classifier is developed by using K-NN, rule-based classifier, SVM, decision trees and logistic regression. The chosen classifiers with linguistic domain features outperformed in comparison to models using the tf-idf features.

Paramesh S.P et al. in [9] explored the effectiveness of Natural Language Processing and various supervised ML techniques to categorize the IT infrastructure helpdesk tickets. The proposed automated ticket classifier classifies the tickets into one of the predefined service categories by parsing the unstructured ticket descriptions. SVM, Logistic regression, Naive

Bayes and K-NN algorithms are used to build the help desk ticket classifier. SVM with 89% accuracy outperformed all other chosen classifiers both on the training and test data.

Jian Xu et al. [10] 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 by signature-based ticket 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 generates various ticket groups and corresponding signatures. A classification algorithm then determines the label of unseen data by finding the most similar group signature.

Roy et al. [11] proposes an incident classifier model based on the unsupervised machine learning techniques like clustering to cluster the incident tickets using the prior user's ticket description. The proposed method uses k-means clustering based on a new distance metric which uses the combination of Jaccard distance and cosine distance for fixed and free fields of the tickets respectively. The clustering is then followed by labelling by extracting logical item sets for each of the clusters.

3. PROPOSED METHODOLOGY

Classification of IT Incidents is an instance of text document classification problem with the incident descriptions and its associated category representing the text document and document label respectively. The end-to-end solution diagram for training the proposed IT incident classifier using a LSTM-RNN based deep learning model is depicted in Fig.1.

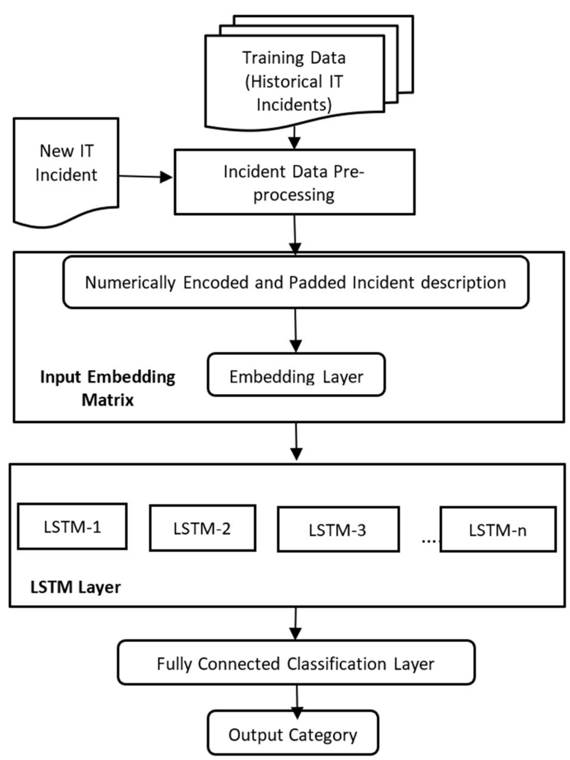

Fig. 1. Proposed solution diagram for IT Incident classification using LSTM neural networks

The key components involved in the development of the proposed incident classification using LSTM neural networks are explained as follows.

3.1 Training data

The proposed Incident classifier can be modelled using a historical incident dataset containing the unstructured IT incident descriptions and the corresponding class labels. In this research work, A real-time IT infrastructure service incidents of a reputed organization containing multiple classes is used as a training data. The incident dataset may contain attributes like incident category, submitter name, severity, priority, description etc but the unstructured incident description and incident category are mainly used to model the proposed classifier.

3.2 Pre-Processing of the Incident Descriptions

The natural language incident descriptions logged by the user are unstructured in nature and generally contains lot of undesired and irrelevant data in the submitted descriptions [10]. The analysis of the incident descriptions of the training data used in this research work revealed that the chosen dataset had lot of irrelevant entities like stop words, special characters, numbers, email ids, date and time, names etc. Pre-processing involves removal of noisy and other unwanted data from initial raw incident data and it is one of the important phases in ML process to build the efficient and accurate ML models. Appropriate NLP techniques are applied to extract and eliminate all irrelevant and unwanted features from the incident descriptions of the training dataset.

Initially, each natural language incident descriptions of the training data are changed into lower case and further tokenized into words, special characters, numbers etc. Frequently used English words i.e., stop words are eliminated as they do not contribute to build an efficient model. Appropriate pattern recognizers are designed to remove the features like numbers, date and time, special characters, emails etc. Part of Speech (PoS) tagging is used to eliminate the functional words such as pronouns, conjunctions, determiners etc. At last, stemming is performed which reduces various derived features to its root form [12]. Pre-processing results in reduced and relevant feature set that can be used to build efficient incident classifier models.

3.3 Numerical encoding and Padding of Incident descriptions

LSTM neural network model accepts only fixed length numerical input data. Each pre-processed incident descriptions are tokenized into words followed by integer encoding of the words based on the word index in the entire corpus of training data. The encoded incident descriptions are then padded using padding mechanism to ensure every input is equal in size. In this research work, we fixed each incident description to contain a maximum of n=120 words and this length 'n' is determined by finding the incident of the training data containing the maximum number of features.

3.4 LSTM-RNN based Deep learning model for IT incident classification

LSTM model for the incident classification mainly consists of Embedding layer which creates the word embeddings representation of the input and are given as an input to the LSTM layer. The LSTM layer learns the long-term contextual dependencies exists among the features. The representation from the LSTM layer is then given as an input to the fully connected classification layer to output the predictions. Each of the network layers involved in the LSTM model for proposed incident classification are explained as follows.

3.4.1 Embedding Layer

Numerically encoded and padded input sentence (i.e., an incident text description in our case) is fed as an input to the embedding layer. Let an incident 'S' consisting of 'n' words be the input to the model. Let $x_i \in \mathbb{R}^d$ be the d-dimentional real valued feature vector of the ith token of the sentence and is represented using the following Eq. (1)

$$x_i = [r_1, r_2, \ldots\ldots r_d] \tag{1}$$

In this research work, a word embedding vector dimension of $d = 100$ is used 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 an embedding 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 ith token in the sentence S and is represented using the below Eq. (2)

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

Here, $\oplus$ is the concatenation symbol. So, the dimension of our input embedding matrix is $n \times d$ = 120x100.

3.4.2 LSTM Layer

LSTM layer is a neural network layer where the regular network units are replaced with a repeating LSTM block in the form of chain like structure as depicted in the below Fig.2.

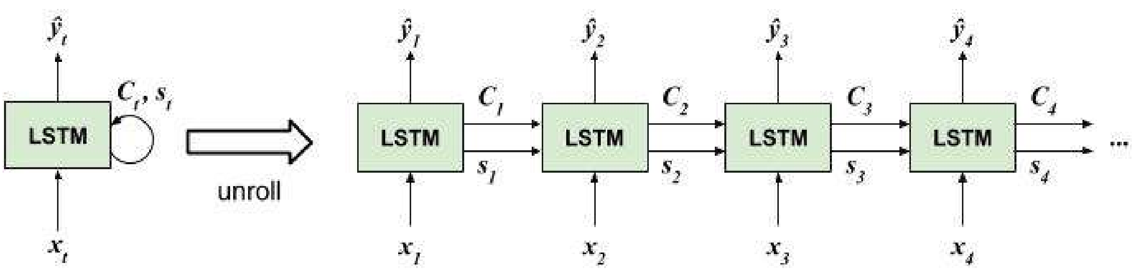

Fig.2. LSTM layer with repeating network units

Here, x_t is the input word at the time 't' and y_t is the hidden state output.

Each LSTM blocks contains 4 different interacting layers in the form of input gates, forget gates and output gates [15,17]. They also have a memory unit (cell state) to store information and this state is controlled by LSTM gates. The LSTM repeating unit at the time-step 't' is shown in Fig.3.

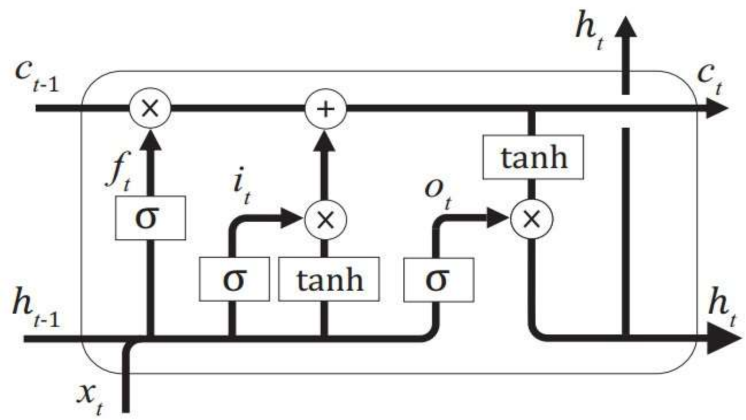

Fig.3. A LSTM repeating unit at the 't' time-step

Here,

- x_t = Input vector at the t-time.
- h_{t-1}= Previous Hidden state.
- C_{t-1} = Previous Memory state.
- h_t = Current Hidden state.
- C_t = Current Memory (cell) state.
- [*] = Elementwise multiplication operation.
- [+] = addition operation.

The input of each LSTM unit is x_t (current input), h_{t-1}, and C_{t-1}. Similarly, the output is h_t and C_t. The main component of LSTM is cell state or memory state C_t, the top horizontal line indicated in the Fig.3. The LSTM units have the ability to add or erase the information to the memory state controlled by gates. Each LSTM gate is a neural net layer with sigmoid activation function and an elementwise multiplication operation. The sigmoid layer outputs numbers between 0 and 1 indicating how much information to pass. A '0 and '1' respectively means allow nothing and allow everything through.

Sigmoidal forget gate layer f_t decides what information to be ignored or erased from the cell state. It takes the h_{t-1} and x_t as inputs and results a number between 0 and 1 for each number in the cell state C_{t-1}. A '1' represents keep the data while a 0 represents delete the data. The transition function of forget gate operation is given below in Eq. (3).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

Here, W_f and b_f respectively represents the weights and bias associated with forget gate.

The input gate layer decides what information to be saved in the cell state and this contains two layers. First input gate layer is a sigmoid layer that decides what values to update and second layer is a tanh layer which creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state. Both these layers outputs are added to create new update to cell state. The transition functions of the both the input gates are given below in Eq. (4) and Eq. (5)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

Here, W_i, b_i and W_C, b_C respectively represents the weights and biases associated with the input and candidate layer.

Now the new value of the cell state C_t is obtained multiplying the old state by f_t and then adding $i_t * \tilde{C}_t$ and is mathematically represented using the below Eq. (6).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

The final output of the LSTM unit 'h_t' i.e., hidden state output at time 't' will be based on the filtered version of the content of the cell state. Sigmoidal layer o_t determines what contents of the cell state to output. The cell state is put through tanh (to push the values to be between −1 and 1) and multiplied by the output of the sigmoid output gate o_t. The transition function of o_t and the output h_t is given using the below equations Eq. (7) and Eq. (8) respectively.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

3.4.3 Fully Connected output classification layer

The hidden state output 'h_t' of the last time step of LSTM is the representation of the input instance and it is passed to the fully connected output classification layer. The dimension of the last fully connected output layer is equal to the number of classes in the dataset. Since the current research work is based on the multi class problems, the probability of the input incident belonging to each category is calculated using the softmax activation function and the entire network is trained by minimizing the categorical cross-entropy loss function. In this work, popular Adam optimizer and backpropagation learning algorithm is used for updating the LSTM network parameters (ie, weights and biases) during each training iteration. In each training epoch, batch_size (the number of samples to be considered) is used to compute the loss and the network parameters like weights and biases are updated once based on this value. Dropout regularization could be used in the LSTM network to avoid the overfitting of the LSTM training model [18].

3.5 Performance Evaluation of the proposed LSTM-RNN model

The performance of the proposed LSTM-RNN based incident classifier is evaluated using various classifier evaluation measures like accuracy, precision, recall and f-score and these metrics are described as follows.

Consider an IT incident classification with binary class problem having only positive and negative incident categories. Let TP is the positive instances that were classified correctly, FP is the positive instances that were incorrectly classified, TN is the correctly classified negative incidents and FN is the incorrectly predicted negative incidents.

Accuracy is the number of correct predictions out of the total predictions made by the classifier and is mathematically given using Eq. (9).

$$Accuracy = \frac{TP+}{TP+TN+FN+FP} \tag{9}$$

Precision evaluates the number of correctly classified positive instances among all the positive instances predicted by the incident classifier and is given by Eq. (10)

$$precision = \frac{TP}{TP+F} \tag{10}$$

Recall is the fraction of relevant positive incidents that are correctly predicted by the proposed classifier and is calculated using Eq. (11)

$$recall = \frac{TP}{TP+FN} \tag{11}$$

Since our research work uses a dataset containing multiple incident categories, precision and recall metrics are computed for each incidents label and then these values are averaged to get the final result. The precision and recall scores can be combined into another metric called f-score which expresses both properties and is calculated using the below Eq. (12)

$$f - score = \frac{(2 \times precision \times recall)}{(precision + rec \quad)} \tag{12}$$

In order to check the efficacy of proposed LSTM-RNN based IT incident classifier, the performance of proposed model is compared with the traditional ML algorithms like SVM, Naive Bayes, Logistic regression and KNN. After the comparative study of all the chosen classifiers, the incident classifier model which outperforms all other chosen classifiers in terms of accuracy and other evaluation metrics is selected as the best trained incident classifier. The best trained model can be further used for making prediction on the new or unseen incidents.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

The computing platform used for this research work consists of 11th generation Intel core i5 processor, 8GB SD RAM, 512GB SSD, NVIDIA GeForce MX450 Graphics card and Windows 10 with 64-bit Operating System. The experimental results obtained at various phases of the development of proposed incident classifier using LSTM-RNN deep neural networks are discussed below.

4.1 Incident data collection and Pre-processing

A real-world IT infrastructure incident dataset containing incident description and its associated category is used for the experimental purposes. Hardware problems, Printer, Skype issue, email related issues, network issues etc. are some of the categories associated with this dataset. The snapshot of the sample chosen infrastructure incident dataset is given below in Fig.4.

Description	Classification	Business Service	Priority	Impact
Hi, Access control desktop is responding to slow, kindly Ins	Hardware Problem	Desktop	3	4
How to install Focal Point Trial version software which is f	Project Specific Software	Service Request	3	4
VPN Installation	VPN issue	Desktop	3	4
laptop crashed	Hardware Problem	Desktop	3	4
My Avaya device is not working properly please configure	Hardware Problem	Laptop	3	4
mouse to be configured	Hardware Problem	Desktop	3	4
Install forticlient software on my desktop.	VPN issue	Laptop	3	4
Unable to connect below server. IP:10.234.12.54	Hardware Problem	Service Request	3	4
I want to install QGIS software for project purpose in my s	Project Specific Software	Desktop	3	4

Fig.4. Sample of the chosen IT infrastructure incident dataset

Here, the 'Description' and 'Classification' fields are mainly used for building the automated incident classifier. The exploratory analysis of the chosen raw IT incident dataset revealed the following statistics about the dataset as given in the Table 2.

Table 2. Description of the IT incident dataset

Description	Dataset
Total number of IT incidents in the raw dataset	10742
Number of incident categories in the raw dataset	18

It is found from the initial analysis of the dataset that huge amount of irrelevant and noisy data was present in the natural language incident descriptions. Data cleaning is done to eliminate all the undesired and noisy data. The features count details during the pre-processing phase is given in Table 3.

Table.3. Feature count details at various phases of pre-processing

Description	Feature Count
Distinct features prior to data pre-processing	6302
Distinct features after eliminating the stop words	6228
Distinct features after eliminating all other irrelevant and undesired features	3600

4.2 Numerical Encoding and Padding of IT incidents

Each pre-processed incident descriptions of the training dataset are encoded as integers and are further subjected to padding to ensure each instance are of fixed length i.e., n=120 in our case. The sample output representation screenshot after encoding and pre-padding of each service incidents is shown in Fig.5.

```
[[   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.5. Encoded and padded incident descriptions of the dataset

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 $n \times d$= 120x100. Python's Kera's embedding function is used in this research work to dynamically create the word embedding vector corresponding to each word of the input sentence. The input embedding matrix (120x100) corresponding each input IT incident is then fed to the LSTM-RNN based classifier model as an input.

4.3 Building and Evaluation of the LSTM-RNN based IT Incident classifier

The proposed LSTM-RNN based incident classifier model is trained using randomly chosen 80% of the training dataset and rest of the 20% of data is used for evaluating the model performance. During the modelling of the proposed incident classifier, the choice of hyper parameters of the LSTM model such as number of LSTM units, epochs, learning rate etc. may influence the performance of the incident classifier. Several experiments were conducted to choose the best values of these hyperparameters. In this research work, the optimal hyperparameters used to build the proposed LSTM based incident classifier is shown in Table.4.

Table.4. Optimal hyperparameters of LSTM model

Hyperparameter	**Choice**
Word Embedding size for each feature	100
Number of LSTM units	100
learning rate	0.001
loss function	categorical_crossentropy
Optimization algorithm	Adam Optimizer
Epochs	30
Batch size	64

During the training of the proposed LSTM incident classifier with optimal hyper parameters, the loss and accuracy plots of the model between the training and validation set (20% of the training data) is recorded. The following Table.5. shows the performance evaluation results of the proposed LSTM based incident classifier during the training phase.

Table.5. Performance results of LSTM model during training phase

Model	**Training**		**Validation**	
	Accuracy (%)	loss	Accuracy (%)	loss
LSTM	91.0	0.18	90.5	0.37

From the Table.5, it indicates that LSTM based incident classifier performed well with a training and validation accuracy of 91.0% and 90.5% and a quite low training and validation loss of 0.18 and 0.37. The performance of the proposed model using the accuracy and loss function plots at various training epochs between the training and validation set is shown in Fig.6.

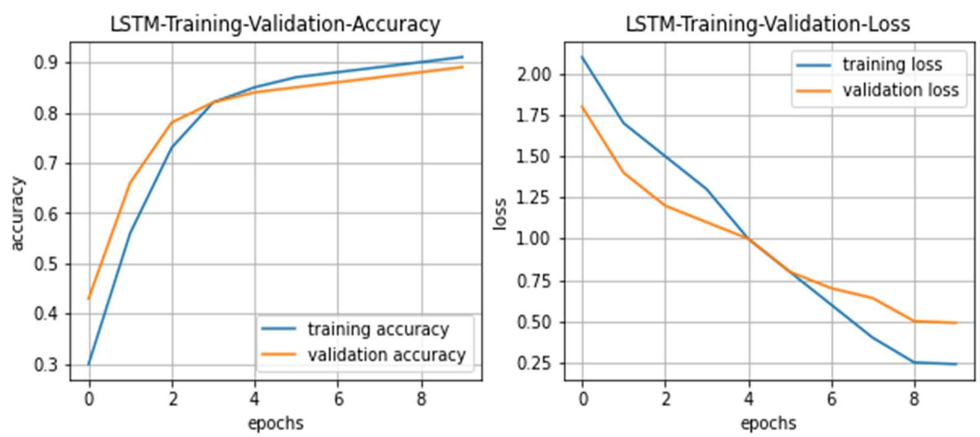

Fig.6. Accuracy and loss plots of LSTM incident classifier during training phase

The proposed LSTM model performance is also validated against the 20% test data using various classification evaluation measures. The LSTM model achieved 90.8% accuracy with a loss of 0.3. The proposed model effectiveness is also compared with traditional machine learning classifiers like Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), Logistic Regression (LR) and K-Nearest Neighbour (K-NN) using the test set and the obtained results are shown in Table.6.

Table.6. Performance comparison of the ML models and the proposed LSTM models

Model	Accuracy	Precision	Recall	F-score
LR	80.67	81.64	80.87	80.53
KNN	68.90	69.75	68.35	68.53
MNB	69.40	72.53	68.40	69.32
SVM	88.20	88.46	87.43	88.06
LSTM	**90.80**	**90.90**	**90.78**	**90.92**

From the Table.6, it indicates that the proposed LSTM based incident classifier with 90.80% accuracy over the test set outperforms the chosen traditional machine learning models in classifying the IT service incidents.

5 CONCLUSION

IT Incident classification and dispatching of incidents to a proper domain team for resolution plays an important role in any kind of service-based systems. Manual classification of incidents in the existing IT service systems may results in misclassification of tickets and results in significant delays in getting the resolution for the problem. To overcome the limitation of manual classification of incidents, an automated IT incident classifier based on deep neural network using LSTM recurrent neural network is proposed in this research work. The proposed system automatically classifies the IT incident into one of the predefined categories by mining the natural language description entered by the user. The methodology uses a combination of word embeddings, LSTM layer and a fully connected output classification layer for the autocategorization of IT incidents. NLP techniques like tokenization, stop words removal, PoS tagging, stemming etc. are leveraged to pre-process the unstructured incident descriptions of the initial raw data. The pre-processed incident descriptions are then represented using the dense word embedding representation. The proposed LSTM model uses this word embedding representation of the input incident and effectively learns the long-term contextual dependencies exists in the incident description to accurately classify the input IT incident to one of the predefined categories.

The proposed LSTM-RNN based incident classifier is validated using a real-world IT infrastructure incident dataset and it achieved a training and validation accuracy of 91% and 90.5% respectively during the training phase. The LSTM model performance is also compared with the classical machine learning models using the test data and the results shows that LSTM based incident classifier achieved a good accuracy of 90.8% in comparison with SVM (88.20%), LR (80.67%), MNB (69.4%) and KNN (68.9%). The proposed approach greatly reduces the manual efforts and human errors by automating the task of service desk agent.

Effective resource utilization, quicker response time, improved end user experience, growth in organization business etc. are some of the other benefits of the proposed automated incident classifier model.

REFERENCES

1. P. Kubiak and S. Rass, "An Overview of Data-Driven Techniques for IT-Service-Management", IEEE Access, Vol.6, pp. 63664-63688, 2018.
2. S.D. Galup, R. Dattero, J.J. Quan, and S. Conger, "An Overview of IT Service Management", Communications of the ACM, Vol. 52(5), pp. 124-127, 2009.
3. Marcu, P., Grabarnik, G., Luan, L., Rosu, D., Shwartz, L. and Ward, C., "Towards an optimized model of incident ticket correlation", In: Proceedings of the IFIP/IEEE International Symposium of Integrated Network Management, (IM), pp. 569-576, 2009.
4. Salah, S., Maciá-Fernández, G., Díaz-Verdejo, J. E., & Sánchez-Casado, L, "A Model for Incident Tickets Correlation in Network Management", Journal of Network and Systems Management, Vol.24(1), pp.57–91, 2015.
5. Mucahit Altintas and Cuneyd Tantug, "Machine Learning Based Ticket Classification in Issue Tracking Systems", Proceedings of International Conference on Artificial Intelligence and Computer Science, pp. 1-6, 2014.
6. Al-hawari, Feras & Barham, Hala, "A Machine Learning Based Help Desk System for IT Service Management", Journal of King Saud University, Vol.33(6), pp.702-718, 2021.
7. N.A. Harun, S.H. Huspi and N.A. Iahad, "Question Classification Framework for Helpdesk Ticketing Support System using Machine Learning", Proceedings of International Conference on Research and Innovation in Information Systems, pp. 1-7, 2021.
8. Revina, K. Buza and V.G. Meister, "IT Ticket Classification: The Simpler, the Better", IEEE Access, Vol. 8, pp. 193380-193395, 2020.
9. S.P. Paramesh and K.S. Shreedhara, "Automated IT Service Desk Systems using Machine Learning Techniques", Proceedings of International Conference on Data Analytics and Learning, pp. 331-346, 2018.
10. J. Xu, H. Zhang, W. Zhou, R. He, and T. Li, "Signature based trouble ticket classification", Future Generation Computer Systems, Vol.78(1), pp.41–58, 2018.
11. S. Roy, D.P. Muni, J.Y.T. Yan, N. Budhiraja and F. Ceiler, "Clustering and Labeling IT Maintenance Tickets", Proceedings of International Conference on Service- Oriented Computing, pp. 829-845, 2016.
12. Ikonomakis, M., Kotsiantis, S. and Tampakas, V, "Text classification using machine learning techniques" WSEAS Trans. Comput, Vol.4(8), pp.966–974, 2005.
13. Marcin Michał Mirończuk and Jarosław Protasiewicz, "A recent overview of the state-of-the-art elements of text classification", Expert Systems with Applications, Vol.106, pp.36-54, 2018.
14. S. P. Paramesh and K. S. Shreedhara, "A Deep learning-based IT service desk ticket classifier using CNN", ICTACT Journal of Soft Computing, Vol. 13, No. 1, pp. 2805-2812, 2021.

15. T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]", IEEE Computational Intelligence Magazine, Vol.13(3), pp.55-75, 2018.
16. S.Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, Vol. 9(8), pp. 1735– 1780, 1997.
17. Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao, "Deep Learning Based Text Classification: A Comprehensive Review". ACM Computing Surveys, Vol.54(3), pp.1-40, 2021.
18. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research, Vol.15(56), pp.1929−1958, 2014.