Journal of Harbin Engineering University
ISSN: 1006-7043
Vol 45 No. 6
June 2024

Artificial Intelligence Classification for IT Ticketing Data Using Improved Feature Analysis Techniques

K. Durga Bhavani[1], T. Rajasri[2], N. Ramadevi[3], T.Srinivasa Rao[4], P. Udayaraju[5] , T.Venkata Narayana[6]
[14,5,] Department of CSE, SRKR Engineering College, Affiliated to JNTUK, Bhimavaram, AP, India
[2,3] Department of IT, SRKR Engineering College, Bhimavaram, AP, India.

Abstract— IT ticketing services are potentially increasing across many corporations in today's internet world. Therefore, the automatic classification of IT tickets becomes a significant challenge. Feature selection becomes most important, particularly in data sets with several variables and features. However, enhance classification's precision and performance by stopping insignificant variables. This Automation in unsupervised ticket classification is a massive impediment to improving the IT support systems. This article the classification of different IT tickets. Through our earlier research, we have categorized the unsupervised ticket dataset. As a result, we have converted the dataset into a supervised dataset. Machine learning algorithms such as Support Vector Machine (SVM), Gaussian Naïve Bayesian, Decision Trees, logistic regression, KNN, and CNN were used. In addition, we have used Feature ranking and feature selection techniques to improve the efficiency of Machine Learning algorithms. However, compared to the ML algorithms, the DL algorithms, like the CNN algorithm, provide a better classification of the token IDs and better accuracy, which is discussed in detail in the results and discussion.
Keywords: Machine Learning, Incident Response, Text Mining, insert, Support Vector Machine

1. INTRODUCTION

While the global economy has focused on services rather than products, technological advancements have kept pace. Because of the wide range of electronic platforms that offer services, information technology has become a vital part of our daily lives. Many people use them for leisure, shopping, and other activities. Every company now has a collection of applications that have evolved due to digitization. A large and complicated IT infrastructure is needed to support this product line. These advances demonstrate that IT support systems are critical in an organization's support operations [2]. In contrast, huge organizations spend millions of dollars on commercial text classification algorithms for small enterprises. These procedures are usually elaborate, one-size-fits-all programs that emphasize accuracy at the expense of speed.

IT workers face various difficulties, including building and infrastructure, software, and HR issues. On the Internet, the helpdesk is used by employees of an organization to report an issue. The tickets are issued by a similar intelligent group or service desk employee based on the ticket category. Ticket categories, priority, and severity are just a few of the structured fields in web-based IT service desk solutions. A free-form field called "ticket description" allows the user to submit a ticket description in their language. Employees manually select the

Problem's category, priority, and severity, as well as its description in standard English, while creating trouble tickets. Manual selection of the ticket category by the end user may lead to an incorrect ticket classification because it is based on the user's impression of the problem and if the user has registered the issue in the relevant category. When tickets are incorrectly categorized, they are sent to the wrong resolution group, which causes a delay in resolving the issue tickets. Conventional service desk systems work best with well-structured datasets.

We can use various machine-learning approaches to build an automatic ticket classification system that addresses these issues. For example, to categorize a service desk ticket, an automated ticket classifier analyses the ticket's end-description user in natural language, using supervised and unsupervised machine learning approaches to build ticket classifier models. Furthermore, classifier models are created using supervised machine learning techniques, such as classification algorithms, if the label of analytical ticket data [2, 6, 8, 9, 10]. Therefore, this paper proposes a machine-learning-based classification of IT tickets.

2. Related Work

This section provides detailed information on various earlier research works focused on IT-Ticketing. Some of the earlier research work can do perfect analysis only on text data and is unsuitable for alpha-numeric data. Some recent research has stated that machine learning models are suitable for data analysis. One author in [2] discussed a Ticker classifier system for the different service desks. The core data is collected from the IT infrastructure helpdesk and applied NLP processes, like pre-processing and normalization. Different strategies are applied for solving data-related problems. It has aimed to deal with undesirable, imbalanced, and correcting wrongly labeled data. The author focused on improving IT services using data mining and machine learning models to overcome research issues. In [3], the author proposed a new IT model used to solve several research issues automatically. The proposed model extracts accurate knowledge based on the problems identified by previous clients and gives better solutions. The proposed approach also focused on providing automated services and reducing human effort. In [4], the author has developed a hierarchical model that utilizes the classification based on ticket analysis. The proposed model utilizes the Gabour model to find the optimal classification technique to configure every ticket. The mining technique in the proposed model works like an expert system and gives an accurate analysis.

The authors in [5] presented a method for categorizing and grouping incident tickets under the user-provided ticket description. The ticket grouping is done by clustering method using unsupervised machine learning approaches. The output of the proposed model is measured by using the Jaccard and cosine distances to confirm the clustering accuracy. Finally, the K-means clustering method is used to improve the clustering accuracy. The authors in [6] proposed a new model that combines SVM to classify the tickets based on events related to previous data. Compared with existing models, the proposed SVM model achieves better accuracy. In [7], the probabilistic concept model is proposed to analyze the support desk tickets. The proposed model analyzes all the keys considering topic modeling, clustering, and IR method. This method first groups the tickets based on the concepts extracted from the tickets' descriptions written in phrase form. Bluefin is a ticket classifier model proposed in [8] to classify unstructured problem tickets.

In this method, events' data and the associated tickets are correlated for context-based classification. Bluefin's performance is compared to Smart Dispatch, an SVM-based ticket classification tool. The authors in [9] identified a way of classifying Change Requests (CR) as one of the activities that might be found in a catalog. It uses an Information Retrieval (IR) technique founded on Lucene, a text search engine available as open-source software, and supervised machine learning algorithms for change request classification. The performance of techniques based on machine learning is superior to that of the IR method; nevertheless, these techniques require a significant quantity of training data. In [10], an automated classifier called an intelligent application is developed. The automated classifier integrated with SVM and differential word weighting method. The technology can categorize tickets by mining past user ticket descriptions and linked labels. The authors in [11] provide NetSieve, which analyses network issue complaints based on the user's ticket description to solve issues based on symptoms; all the activities are troubleshooting and redesigning tasks. The proposed approach solves the problems in the network. To accomplish these objectives, NetSieve combines techniques from natural language processing with the initialization of knowledge and modeling of ontology.

In [12], the authors have proposed a new model that analyzes the issues from ticket descriptions. The application of data retrieval methods and domain ability is needed to extract the points from the problem-based tickets. Various clustering techniques remove the issues from automated and manual tickets. A new classification model has been proposed in [13] that combines the text mining methods like ensemble algorithms, ER models, and recount of frequency to solve issues. The authors in [14] have developed Trouble Miner to sort trouble tickets according to their underlying causes. According to the results, the majority of tickets are caused by disruptions in the network cables and routers [14]. In [15], the authors constructed regression and classification models to predict the resolution times. It was decided to eliminate the fields that held text data because the text had to be entered by a human every time. A text area is included in this work; however, the reader is not produced manually but rather by a machine. Because of this, the text field is mined for helpful information.

Regarding classification, the resolution time was divided into three categories, and the resulting model had an accuracy of approximately 74.5 percent. On the other hand, when it came to regression, the artificial neural network had the lowest MAE, which was 24.8 hours [16]. The authors in [17] have employed data mining and machine learning methods to determine the underlying cause of network issues. It allows him to provide engineers with actionable recommendations and, as a result, reduces the amount of time spent on troubleshooting. The model had an accuracy of up to 90 percent when predicting the root cause of the most prevalent root cause and only 70 percent when discriminating between up to 20 different root causes [17, 18, 19]. The authors in [20] proposed a new model for detecting abnormal eye diseases such as ARDM using DL models.

3. Proposed Approach

The current dataset is unlabeled and unsupervised. The unsupervised ticket dataset has been categorized and labeled for fast transformation. The below diagram presents a pictorial representation of our approach. In the obtained dataset, the body attribute is textual, whereas all the others are numerical. Hence, we have carried out different types of performance analysis. Applying the predictive models, feature Ranking, and selection techniques to the dataset with and without the body attribute. Pre-processing was carried out to convert the textual data in the body tag to numerical data to use the body attribute in the dataset. To perform this, we have to use the count vectorizer library. After the conversion is done, one more pre-processing step is carried out, i.e., to normalize all the data into a range of 0 to 1. After this step, feature ranking is carried out to understand which features are of utmost importance, and the feature selection technique is used to improve the efficiency of the predictive models such as the Support Vector Machine Classifier (SVM/SVC), Gaussian Naive Bayesian, Decision Trees, logistical regression, and KNN.

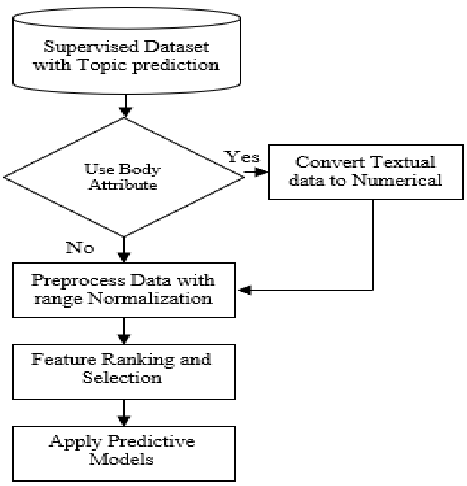

Fig. 1: Proposed approach

3.1. Dataset

This dataset was retrieved after performing clustering and labeling mechanisms obtained from our previous study. The best resultant algorithm of the prior survey, LDA-based Topic Prediction, which contains the 13 topics, was used as the target attribute for classification. The dataset includes 47664 incidents initially taken from the Service Now platform. Table 1 shows how the characteristics in the dataset were used to perform this study. Yes/No values in the usage column suggest using a particular feature for this study.

3.2. Environment

3.3. Logistic Regression

3.4. SVM

Support vectors are the values that are situated in the closest proximity to the categorization boundary. Support vector machines, often known as SVMs, are a common classification and regression analysis approach utilizing data analysis and pattern recognition. A more realistic description of a support vector machine would specify that it constructs a hyperplane or collection of hyperplanes to categorize all inputs in a space that is either high-dimensional or even infinite. The support vector machine (SVM) aims to achieve a margin that is as large as possible between the hyperplane and the support vectors.

3.5. CNN

It is one of the deep learning algorithms that takes input data and assigns them tags and ids based on their weights or parameters. These tags and ids are used to differentiate the characteristics of the features extracted from the data. It also requires very less pre-processing of the data, as it classifies them by itself and learns from them. The functioning of the CNN algorithm is similar to that of the human brain. It consists of neurons that pass through several networks to modify the extracted data and finally learn the features. It also takes advantage of the spatial and temporal features of the dataset and improves its functioning. Mostly, it is suitable for image datasets as it takes advantage of the image's pixel information. The layers used in the CNN algorithm are discussed in detail in the following table.

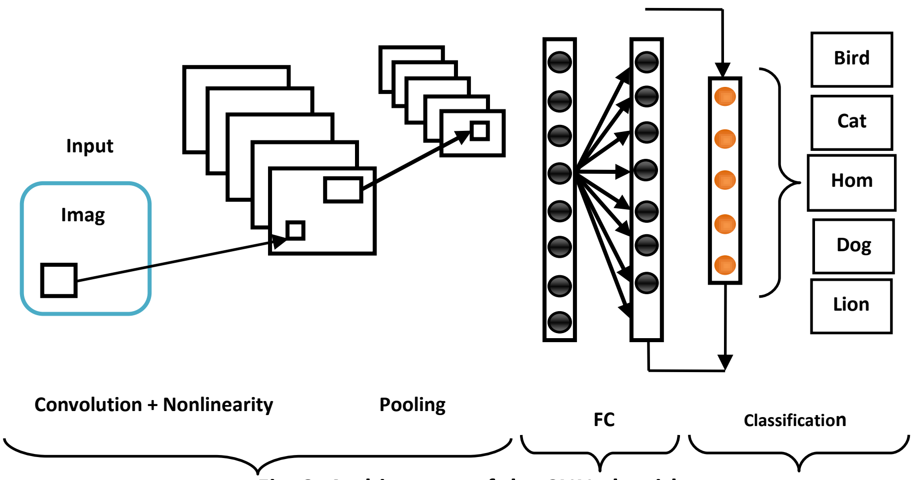

Fig. 2. Architecture of the CNN algorithm

Let the CNN have in m blocks. Let us input x_l to the l^{th} block, and the mapping of the residual is $f_l(.)$, then the x_{l+1} is the output of the l^{th} block is derived as follows

$$x_{\ell+1} = x_\ell + \mathrm{f}_\ell(x_\ell) \qquad (1)$$

Full Self-attention (Full-SA) Network:

The below equation shows that, l^{th} block is denoted as $M(.;w_l)$, and it is placed in w_l parameters. Then the equation is developed $M(f_l(x_l);W_l)$ which includes the processing end of the extraction process. Finally, the residual output is $f_l(x_l)$, as follows. Where $l = 1, \ldots.,$ m and $\odot$ denotes the element-wise multiplication. Equation (2) defines the computation cost, and the parameters are increased based on the number of blocks m.

$$x_{\ell+1} = x_\ell + \mathrm{M}(\mathrm{f}_\ell(x_\ell);\ \mathrm{W}_\ell) \odot \mathrm{f}_\ell(x_\ell), \qquad (2)$$

Connection Scheme:

Assume that the CNN has m blocks. A sequence $a = (a_1, a_2 \ldots, a_m)$ indicates a connection scheme, where $a_i = 1$ and i^{th} block is connected to a SAM. The scheme formulated in equation 3 is given below.

$$x_{\ell+1} = x_\ell + (\mathrm{a}_\ell.\mathrm{M}(\mathrm{f}_\ell(x_\ell);\ \mathrm{W}_\ell) + (1 - \mathrm{a}_\ell).1) \odot \mathrm{f}_\ell(x_\ell), \qquad (3)$$

Al, the one vector is defined here as 1, and the l lies between the 1 and m. The all-one vector is represented as a full-SA network, and the CNN allows the neurons if a represents 0.

A d-dimensional set K is considered in which the T(x) is referred to as the Lipschitz continuous and Lebesgue integrable function. The CNN structure consists of $R_{full}(x, \theta_{full})$. The constant in this network is $\epsilon > 0$. $\int_K |R_{full}(x, \theta_{full}^0) - T| dx < \epsilon_0/2$. All the layers have a width of $R_{full}(x, \theta_{full}^0)$ which is greater than d and the depth of the network, which is also represented as $R_{full}(x, \theta_{full}^0)$ that exceeds the value of a constant ϵ_0. The subnetwork is defined as $R_{\text{full}}(x)$ in which the $R_{full}(x, \theta_{full})$ which can be written in the following form,

$$\int_k |R_{full}(x, \theta_{full}^0 - R_{sub}(x)| dx \leq \epsilon. \quad (4)$$

The connection frequencies between the networks are measured through a connection score of the scheme set. Let the number of blocks in the network be m and the connections schemes are represented as N. For every scheme, $a_i = (a_{i1}, a_{i2}, \ldots, a_{\mathfrak{J}}), a_{ij} \epsilon\ 0, 1$

$I = 1, \ldots, N, j = 1, \ldots, m$, and the connection score is defined in the following manner,

$$\left(\frac{1}{N}\sum_{i=1}^{N} a_{i1}, \frac{1}{N}\sum_{i=1}^{N} a_{i2}, \ldots, \frac{1}{N}\sum_{i=1}^{N} a_{im}\right) \quad (5)$$

The $\chi_\theta(q_0)$ represents the fully connected network, which produces the connection schemes, and the learnable parameters are represented as %theta. The constant vector 0 is represented as q_0. The p_θ is the output of $\chi_\theta(q_0)$. The p_θ is represented as a series $(p_\theta^1, p_\theta^2, \ldots, p_\theta^m)$ and the p_θ^i can be represented as the probability of getting connected to the ith block of the SAM. A controller output helps in sampling the realization a, which is $a \sim p_\theta$. The scheme has the probability of $\hat{p_\theta} = (\hat{p_\theta^1}, \hat{p_\theta^2}, \ldots, \hat{p_\theta^m})$. The G(a) is denoted as the reward is given to a. The controller has the parameter set of θ and the η be the policy gradient of the learning rate.

$$R_\theta = G(a). \sum_{i=1}^{m} \log \hat{p}_\theta^i, \quad \theta \leftarrow \theta + \eta. \nabla R_\theta \quad (6)$$

In this manner, the controller provides the probability for the reward G. Searching for a good structure of G can help find a better one. The better reward G can be found through the connection ratio and accuracy. The sub-network $\omega(x \vee a)$ provides a validation accuracy of g_{val}, which is obtained by sampling the supernet of the reward.

$$\frac{I_t(\text{CNN with SAMs}) - I_t(\text{Original CNN})}{I_t(\text{Original CNN})} \times 100\% \quad (7)$$

The network's inference time $I_t(.)$. The batch size is defined to be between 50 to 1000 times.

The $T(x)$ is defined as the Lipschitz continuous function, and the K is the d-dimensional compact set's lebesgue integrable function. The overall subnetwork consists of the depth and width of the layer $R_{full}(x, \theta_{full})$ which is smaller than the constant ϵ_0.

$$\int_K \left|R_{full}(x, \theta_{full}^0) - R_{sub}(x)\right| dx \leq \in \quad (8)$$

$$\int_{\mathbb{R}^d} |f(x) - R(x)| dx \leq \in \quad (9)$$

$$\int_{\mathbb{R}^d} \left|f(x, \theta_f^0) - T\right| dx \leq \in \quad (10)$$

The skip connections are seen from the formulas, $g(x, \theta_g^0) = f(x, \theta_f^0)$.

$$\int_{\mathbb{R}^d} \left|f(x, \theta_f^0) - T\right| dx = \int_{\mathbb{R}^d} \left|g(x, \theta_g^0) - T\right| dx \leq \in \quad (11)$$

$$\omega_K(r) = \max_{x,y \epsilon K, ||x-y|| \leq r} |f(x) - f(y)| \quad (12)$$

Let T(x) can be seen that Lipschitz continuous function algorithm, in which the following equation shows its functioning,

$$|T(x) - T(y)| \leq L|x - y|, \quad (13)$$

,then we have given that $w_k(r), = Lr = \epsilon / Vol\ (K)$.

$$r = \frac{\in}{Vol(K).L} \quad (14)$$

Here, if the r= %epsilon/ (Vol(k).L) , then the $\epsilon \equiv (\epsilon_{0,1})$,

$$O(1/r^d) = O((\frac{L}{\in})^d < O((\frac{L}{\in_0})^d = C((\frac{L}{\in_0})^d, \quad (15)$$

The constant C in the equation is multiplied by the Lemma value that exists in the CNN $R_{short}(x)$ and its width lies within d and the $C(L/\epsilon_0)^d$ also lies within it.

$$dep(R_{long}) = dep(R_{full}), \quad (16)$$

It can be seen from the equation that the d value is greater that R_{long}. It can also be stated that $x \epsilon K$, and $R_{long}(x) = R_{short}(x)$.

$$\int_K \left|T(x) - R_{long}(x) dx = \int_K \right|T(x) - R_{short}(x) dx \leq \in/2 \quad (17)$$

$$\int_K \left| R_{full}\left(x, \theta_{full}^0\right) - R_{long}(x) \right| dx \leq \int_K \left| T(x) - R_{long}(x) \right| dx \quad (18)$$

$$+\int_K \left| R_{full}\left(x, \theta_{full}^0\right) - T(x) \right| dx \quad (19)$$

$$\leq \epsilon/2 + \epsilon_0/2 \leq \epsilon \quad (20)$$

The R_{long} represents the CNN algorithm that has a width smaller than that of d, while the R_{full} is greater than d. The inequality in the network is satisfied by the R_{full}.

$$m_c^\ell = AVG\left(X_c^\ell\right) = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{chw}^\ell, \quad (21)$$

The features are processed by the sigmoid function $sig(z) = 1/(1 + e^{-z})$. Here the reduction rate is defined as r and the division extracted as '//'. Thus, the size of the hidden layer is C//r. The information obtained from all the channels are fused using the ReLU activation function.

$$[\delta_1; \ldots \delta_C] = sig\left(FC\left(\left[m_1^\ell; \ldots; m_C^\ell\right]; W_\ell\right)\right) \quad (22)$$

The blockwise information of the LSTM is integrated through the EIA module. The average pooling output is termed as m_c^l that is passed to the hidden state h_{l-1}. The zero vectors are termed as the h_0 and c_0.

$$(h_\ell, c_\ell) = LSTM\left(\left[m_1^\ell; \ldots; m_C^\ell\right], h_{\ell-1}, c_{\ell-1}; W\right) \quad (23)$$

The G represents the number of groups, and the feature maps are represented as C//G. The feature maps are grouped as $(C//G) \times H \times W$ that represents the Y^l that lies within X^l.

$$g_c^\ell = AVG\left(Y_c^\ell\right) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} Y_{chw}^\ell \quad (24)$$

The coefficient of importance for each value g in $[g_1^l; \ldots; g_{C//G}^l]$

$$phw = g.Y[:, h, w], \quad (25)$$

The value p_hw is normalized in the following steps,

$$\hat{p}hw = \frac{phw - \mu}{\sigma + \epsilon} \quad (26)$$

The μ and σ^2 are the mean and variance of the tickets which can be calculated through the following equation,

$$\mu = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} phw, \quad \sigma^2 = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (phw - \mu)^2 \quad (27)$$

For the group Y^l, some additional set of parameters (γ, β) are added to rescale and normalize the features, and the attention received by the SGE modules $T[:, h, w]$ are written as

$$sig(\gamma \hat{p}hw + \beta) \quad (28)$$

The full CNN algorithm is accelerated through the g_spa. The g_spa encourages the network to achieve better results by generating fewer connection schemes which is found between the tickets.

$$g_{spa} = 1 - \frac{||a||_0}{m}, \quad (29)$$

More schemes can be explored in this process, as there is mitigation and convergence during the training iterations, and the RND bonus provides better convergence of the iterations. The difference in the output is reduced by the RND().

$$G(a) = \lambda_1 . g_{spa} + \lambda_2 . g_{val} + \lambda_3 . g_{rnd} \quad (30)$$

The proximal policy optimization method is used to speed up the training and sampling of the connection schemes. It also provides better efficiency in utilizing the data, and the tuple is kept in a buffer after updating the parameters %theta and %phi.

$$\kappa = \mathbb{E}_{a \sim P\theta_{old}} \left[G(a) \sum_{i=1}^{m} \frac{\hat{p}_\theta^i}{\hat{p}_{\theta_{old}}^i} \nabla_\theta log \hat{p}_\theta^i \right], \quad (31)$$

$$\theta \leftarrow \theta + \eta . \kappa,$$

TABLE 2. The architecture of the CNN Algorithm

Layer(type)	Output	Shape	Param#
conv2d_1	(Conv2D)	(None, 75, 100, 32)	896
conv2d_2	(Conv2D)	(None, 75, 100, 32)	9248
max_pooling2d_1	(MaxPooling2)	(None, 37, 50, 32)	0
dropout_1	(Dropout)	(None, 37, 50, 32)	0

conv2d_3	(Conv2D)	(None, 37, 50, 64)	18496
conv2d_4	(Conv2D)	(None, 37, 50, 64)	36928
max_pooling2d_2	(MaxPooling2)	(None, 18, 25, 64)	0
dropout_2	(Dropout)	(None, 18, 25, 64)	0
flatten_1	(Flatten)	(None, 28800)	0
dense_1	(Dense)	(None, 128)	3686528
dropout_3	(Dropout)	(None, 128)	0
dense_2	(Dense)	(None, 7)	903
Total params: 3,752,999			
Trainable params: 3,752,999			
Non-trainable params: 0			

The above table shows the convolutional, pooling, and fully convoluted layers used in the algorithm. It also is said that the proposed algorithm provides better ticketing of the IT, and it needs to be confirmed whether the proposed algorithm is more efficient than other existing methods. Most of the existing methods considered classification algorithms, while this paper has considered CNN for efficiently processing the data.

4. Classifiers

4.1. Random Forest (RF)

The fundamental idea underpinning the random forest method is the development of a large number of primary decision trees in the training phase, followed by a vote based on the results across all of the trees in the classification phase. During the training phase, random forests use a technique known as bagging as a general strategy to apply to individual trees inside the ensemble. When using bagging, a random sample from the training set is selected with tereplacement several times, and trees are fitted to these samples. Every tree develops naturally without any intervention from pruning.

4.2. Decision Trees (DT)

Training from decision trees is a kind of supervised machine learning that involves generating a decision tree from a set of training data. Decision tree learning A predictive model known as a decision tree is a projection that moves from observations about an object to predictions about the value it is supposed to have.

4.3. Gaussian Naive Bayes (GNB)

The Gaussian Naive Bayes method is a variation of the Naive Bayes approach that adheres to the Gaussian normal distribution and works with continuous data. It produces a bell-shaped curve when plotted, which is symmetric around the mean of the feature values. In the process referred to as Gaussian Naive Bayes, constant values associated with each feature are believed to follow a Gaussian distribution. Therefore, "Normal distribution" may also refer to a "Gaussian distribution."

4.4. KNN

The k-nearest neighbors (kNN) classification algorithm is simple yet reliable. To classify a new instance, the **k** closest neighbors of the cases are first chosen, and then the main class of those k neighbors is used to determine the type in which the new instance are placed. Therefore, when the kNN technique is used to classify data, the value selected for the parameter k significantly impacts the accuracy of the results.

5. Feature Ranking and Selection

Feature selection is a procedure in which you automatically pick features in your data that contribute the most to the prediction variable or output you are interested in. This selection is made via a process known as feature extraction. The following are some of the advantages that come from completing feature selection before modeling your data:

To avoid overfitting, collect fewer duplicate data. Since there are less data, the algorithms train more quickly. It offers a performance boost model, resulting. In addition, it reduces the amount of time needed for training.

5.1. Chi-Square

Chi-Square feature selection is an example of a feature selection approach often used while working with text data. To be more explicit, we utilize it in the feature selection process to determine whether or not the incidence of a specific word and the occurrence of a particular class are independent of one another. For instance, in statistics, the Chi-

Square test determines whether or not two occurrences may be considered independent.

$\chi 2 = \sum(Oi - Ei)2/Ei$

Oi = Actual Observation

Ei = Expectation.

If the matching Chi-Square score for each feature is high, this suggests that the null hypothesis H0 of independence should be rejected and that the occurrence of the feature and class depend on one another.

5.2. RFE

Recursive feature elimination, often known as RFE, is a technique for selecting features that fit a model and eliminating the part (or features) that are the weakest until the necessary number of features has been attained. RFE tries to remove any dependencies and collinearity present in the model by iteratively deleting a small number of features at each iteration of the loop. The features are prioritized according to the model's coefficient or the feature priority characteristics. RFE necessitates retaining a certain number of features; however, it is not always possible to predict how many elements will be considered legitimate. Therefore, cross-validation is used with RFE to score several feature subsets and choose the collection of features with the highest score. It allows for the optimum features to be determined.

6. Result Analysis

SVM, KNN, CNN, LR, RF, DT are the major classifiers used in the paper to evaluate the performance of the classifiers by comparing their results. The data set is taken from the Kaggle dataset repository and experimented with various machine learning algorithms implemented in Python. Before using the data to train the different models, we had to clean the data and select the most important columns to be included in the model. One of the biggest problems we had with the dataset was that it had many zeros and columns to choose from.

Startup investment can be very risky due to the high failure rate of startups. People like angel investors and venture capitalists have a very high risk while they are investing in startups. To assist startup investors with their decisions, in this project, we aim to find the important features that lead to startup success and forecast a company's success with supervised machine learning methods. One of the execution models that improve employee performance and their process is called IT ticketing. This IT ticketing model is used in this paper and analyzed using a machine learning algorithm can improve the company execution model and success.

We also realized later that the status column had around 80% of the companies as operating status and the rest as closed and acquired companies.

To train the machine learning model, we used investment data about startup companies available on Kaggle. The data has been collected from Crunchbase, a leading website for company insights from early-stage startups to Fortune 1000.

The data had around 54k rows and 39 columns. The dataset had company information such as name, URL, market, country, state, region, city, founded date, first funding date, and last. It also had data on different investment types such as seed, venture equity crowdfunding, undisclosed funding, convertible note, debt financing, angel, grant, private equity, post-IPO equity, post-IPO debt, secondary market, product crowdfunding, and round A-H series funding. Detailed descriptions of the different funding types are available here. The status of the companies was also available and segmented by acquiring, operating, and closing.

Feature Ranking using RFE without Body	**Logistic Regression**	**Random Forest**	**Decision Trees**	**CNN**
Ticket Type	1	5	5	6
Category	2	3	3	5

Performance Measures		LR	RF	DT	CNN
Sub Category 1	3	1	1	1	
Sub Category 2	5	1	1	1	
Business	4	1	1	1	
Urgency	1	2	2	3	
Impact	1	4	4	4	

TABLE 3: FEATURES RANKINGS WITHOUT BODY USING RFE

The category attribute in the dataset consists of 13 categories. All the tickets in the dataset **t** are labeled with topic prediction results from our earlier research work. These tickets are used as input for the predictive models. To analyze the results and performance, we have conducted a detailed study on using the body tag without using the body. Table-1, 2, 3, and 4 show the performance analysis results. Table-1 shows the result of Recursive Feature elimination on the obtained dataset. Results show that Random Forest and Decision tree algorithms ranked 1 for Sub category1, Sub category2, and Business attributes. And Logistic Regression ranked 1 for Urgency, impact, and Ticket type attributes. On the other hand, predictive algorithms such as SVC, Gaussian Naïve Bayesian, KNN, and CNN algorithms were not applicable with RFE and hence denoted as NA.

Body attribute has 10 details when they are converted from textual to numerical value. The results of using RFE with the body attribute are presented in Table 2. When the boy attribute is used, we can observe a change in the ranks of features in the dataset. The Random Forest and decision trees have been awarded a ranking of 2 to 6 for all 10 body attributes. Random Forests and Decision trees had a similar order for the other features. Logistic regression has awarded a rank1 to the body attribute.

Feature Ranking using RFE With Body	Logistic Regression	Random Forest	Decision Trees	CNN
Ticket Type	2	8	8	7
Category	5	1	1	1
Subcategory 1	6	1	1	1
Subcategory 2	8	1	1	1
Business	7	1	1	1
Urgency	4	1	1	1
Impact	3	7	7	6
Body	1	(2-6)	(2-6)	(1-5)

TABLE 4: FEATURES RANKINGS WITH BODY USING RFE

Accuracy	Without Body	87.35	85.41	85.42	98.43
Specificity		95.45	94.56	94.83	97.56
Sensitivity		98.45	97.12	97.34	98.56
Accuracy	With Body	81.03	80.86	82.14	98.32
Specificity		91.33	90.77	92.07	98.56
Sensitivity		95.43	94.58	96.09	98.45

Also, the performance of RFE with Feature ranking with and without the body attributes is presented in Table-4 and 5. Decision trees had a higher Accuracy and better specificity and sensitivity when compared with the Logistic regression and Random Forest while using the Body attribute without the body attribute.

TABLE 5: PERFORMANCE OF RFE FOR FEATURE RANKING

Without Body			
	Accuracy	**Specificity**	**Sensitivity**
Logistic Regression	87.45	95.65	98.51
SVC	87.36	95.44	98.50
Random Forest	85.26	94.43	97.21
Decision Trees	85.41	94.88	97.56
Gaussian	87.03	95.03	98.04
KNN	89.22	96.55	98.91
CNN	89.43	94.67	97.45
With Body			
	Accuracy	**Specificity**	**Sensitivity**
Logistic Regression	82.52	92.74	96.31
SVC	86.86	94.91	97.1
Random Forest	80.81	90.79	94.65
Decision Trees	81.61	91.56	95.88
Gaussian	82.03	92.15	96.23
KNN	83.55	93.27	96.59
CNN	98.55	93.25	97.45

TABLE 6: PERFORMANCE OF CHI-SQUARE ON WITH AND WITHOUT BODY ATTRIBUTE

We have carried out the Chi-Square feature selection technique on our obtained dataset. The number of features value is set to 3. Feature selection has produced a similar result as the Feature ranking, where features Sub Category-1, Subcategory-2, and Business had better results when compared with any other combination of features.

Table 5 shows the performance analysis results of employing the chi-square feature selection technique on the predictive models. KNN has a better accuracy of 89.22% over the other models when body attribute is not used, and SVM/SVC had a better accuracy of 86.86% compared with the others. Below figure-2 shows the mean performance of these models.

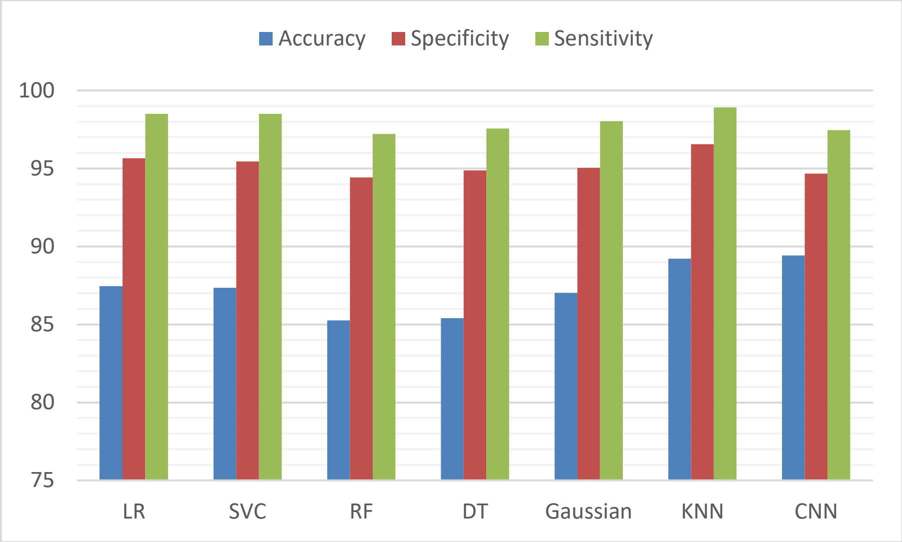

Fig 3. Mean performance of Predictive Models with Chi-Square Without Body Attribute

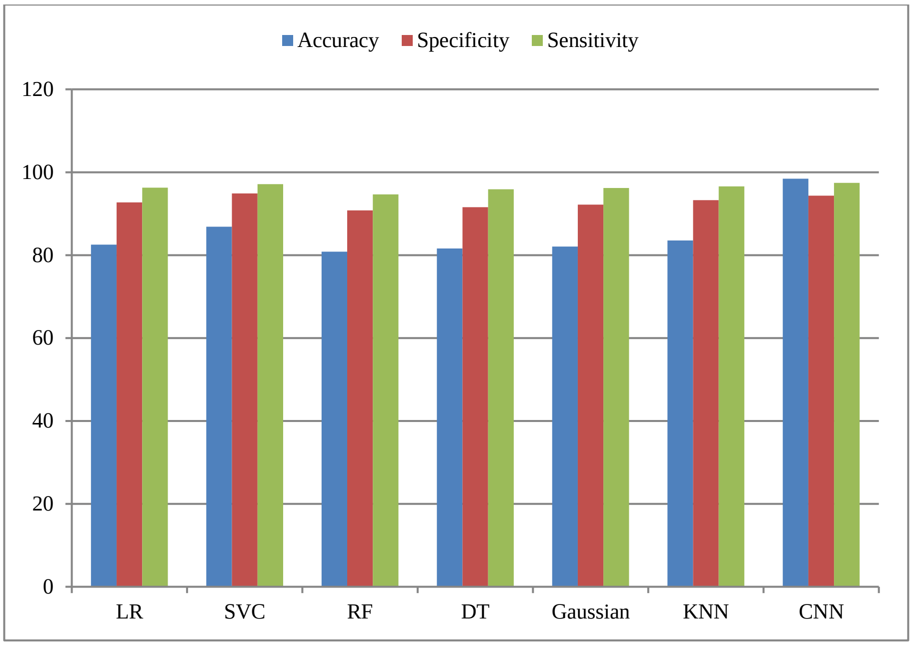

Fig 4. Mean performance of Predictive Models with Chi-Square on With Body Attribute

To evaluate the overall better predictive model, we have conducted a mean performance analysis where the average is calculated considering accuracy, specificity, and sensitivity for both with and without body attributes. The Random Forest algorithm has the lowest mean overall performance at 90.53%, whereas the Support Vector Machine Classifier (SVM/SVC) has the highest performance at 93.36%. The Random Forest algorithm has the lowest mean accuracy at 83.035%, and SVC has the highest mean accuracy at 87.11%.

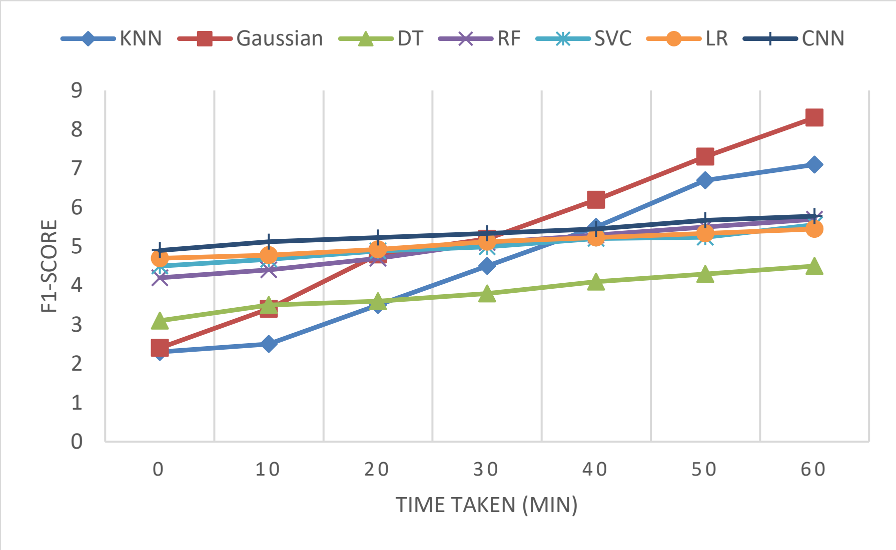

Fig. 5. Delay time for the algorithms

The figure-4 shows the time consumed by the algorithms considered in this work. Based on the time consumed, the F1 score varies, and the maximum F1 score is achieved for the maximum time. In this work, ID management and token ID verification play a crucial role, and the proposed CNN algorithm outperforms all the algorithms computing faster with better accuracy and F1 score.

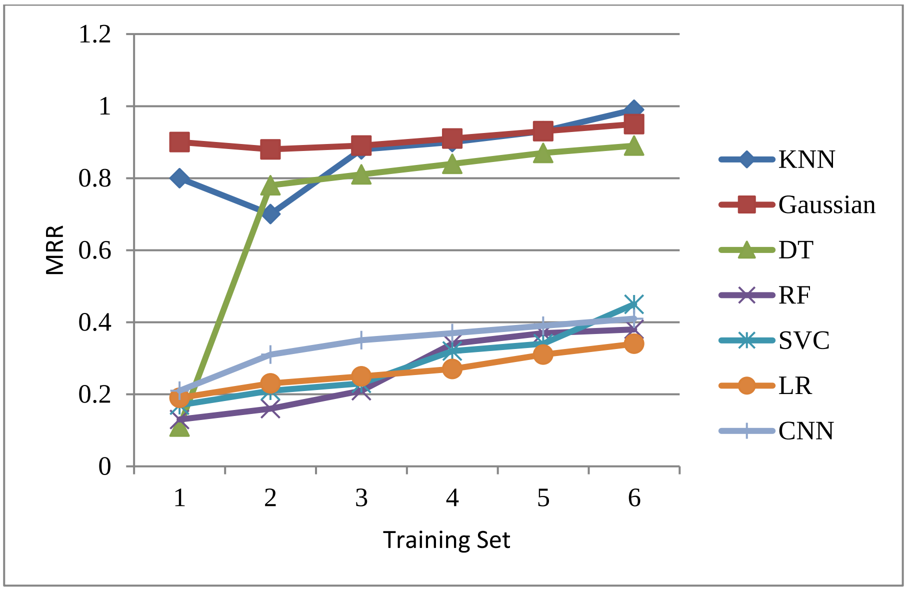

Fig. 6. MRR for the training dataset

All the algorithms are trained with the training dataset to improve the performance of the algorithms shown in figure-5. However, the clustering algorithms provide much less MRR (Monthly recurring revenue) than the CNN algorithm. It also is termed as the inefficiency of the algorithm to learn the features, and it takes time and quality data to improve the accuracy of the algorithms further. But the CNN algorithm with the minimum number of datasets and features provides better classification and MRR.

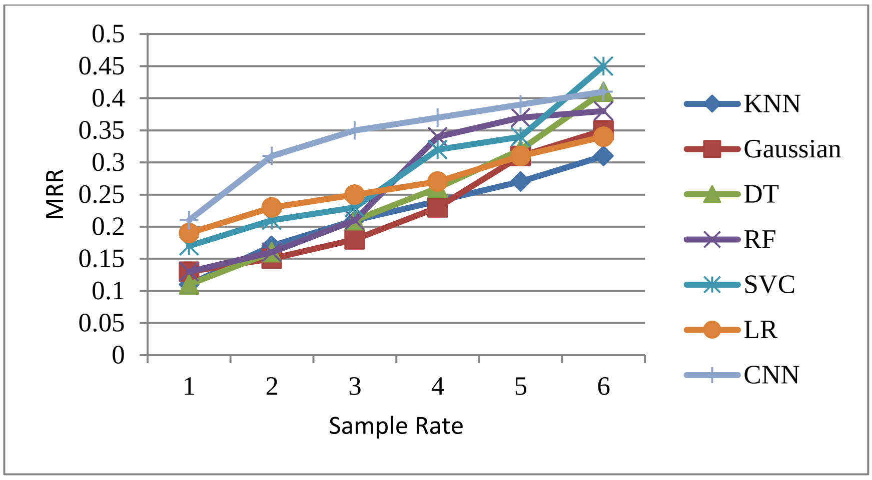

Fig. 7. Mean performance of Predictive Models with Chi-Square

The figure-6 shows the sample rate tuning of the algorithms considered in this work. The CNN algorithm has the minimum sample rate tuning compared to other clustering and regressive algorithms. Through this,, it can be concluded that the CNN algorithm provides efficient and accurate results compared to other algorithms and outperforms other algorithms in terms of cost, resources, and other metrics.

Industry, continent, and total investment are important features. We received the best result when we used SVM and Random Forest for Multi-Class Classification—received the best result when we used Random Forest for Binomial Classification. For future scope, we would like more data for closed and acquired companies, a test model with one-hot encoding, a test with other models like Naive Bayes and XG Boost, test with KNN and SVM on Binary Classification Model. Using Crunchbase API, we can also make a real-time dashboard and deploy a model so that it can assist investors and founders.

7. Conclusion

As a result of our past work, the unsupervised ticket dataset has been classified and labeled, transforming it into a supervised dataset. In the retrieved dataset, only the body attribute is textual. Through this research, we have conducted performance analysis of several feature ranking and feature selection techniques (RFE and Chi-square) combined with predictive models such as SVM/SVC, Gaussian Naive Bayesian, Decision Trees, logistical regression, and KNN. For Feature ranking, the Decision tree algorithm performed better when compared with the Random Forest or Logistic Regression algorithms. KNN algorithm performed well without using textual data when combined with chi-square. While analyzing the overall performance of predictive models (with and without body attributes), when paired with the Chi-Square feature selection technique, the CNN algorithm outperformed all other methods with a mean accuracy of 87.11%.

Author's Contributions: All authors contributed equally.

Funding: No funding was received to assist with preparing this manuscript.

Availability of data and materials: Not applicable.

Declarations:

Conflict of Interest: Not applicable.

Ethical approval: The authors declare that they have no conflict of interest.

References

[1] Weblink to ServiceNow Table API last accessed on 5-July-2022 https://developer.servicenow.com/dev.do#!/reference/api/sandiego/rest/c_TableAPI.

[2] Paramesh S.P., Shreedhara K.S. Automated IT Service Desk Systems Using Machine Learning Techniques. In: International conference on Data Analytics and Learning

2018. LNNS,vol 43,pp.331-346. Springer, Singapore. (2018).

[3] S. Agarwal, V. Aggarwal, A. R. Akula, G. B. Dasgupta and G. Sridhara, "Automatic problem extraction and analysis from unstructured text in IT tickets," in IBM Journal of Research and Development, vol. 61, no. 1, pp. 4:41-4:52, 1 Jan.-Feb. 2017.

[4] C. Zeng, W. Zhou, T. Li, L. Shwartz, G.Y. Grabarnik, Knowledge guided hierarchical multi-label classification over ticket data, IEEE Trans. Netw. Serv. Manag.14 (2) (2017) 246–260.

[5] S. Roy, D. P. Muni, J. Y. T. Yan, N. Budhiraja, F. Ceiler, "Clustering and labeling IT maintenance tickets", 14th ICSOC'16 Banff, pp. 829- 845, 2016.

[6] J. Xu, L. Tang, T. Li, System situation ticket identification using SVMs ensemble, Expert Syst. Appl. 60 (2016) 130–140.

[7] E.E. Jan, K.Y. Chen, T. Idé, Probabilistic text analytics framework for information technology service desk tickets, in 2015 IFIP/IEEE International Symposium on Integrated Network Management, (IM), IEEE, 2015, pp. 870–873.

[8] Gargi .B. Dasgupta, T.K. Nayak, A.R. Akula, S. Agarwal, S.J. Nadgowda, Towards auto-remediation in services delivery: context-based classification of noisy and unstructured tickets, Proceedings of International Conference on Service-oriented Computing (SOC), 2014, pp. 478– 485.

[9] C. Kadar, D. Wiesmann, J. Iria, D. Husemann, and M. Lucic, "Automatic classification of change requests for improved IT service quality," Proc. of SRII Global Conference, pp. 430–439, 2011.

[10] S. Agarwal, R. Sindhgatta, B. Sengupta, SmartDispatch: enabling efficient ticket dispatch in an IT service environment, ACM SIGKDD (2012) 1393–1401.

[11] R. Potharaju, N. Jain, C. Nita-Rotaru, Juggling the jigsaw: Towards automated problem inference from network trouble tickets, in: Presented as Part of the 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI 13, 2013, pp. 127–141.

[12] V. Shimpi, M. Natu, V. Sadaphal, et al., Problem identification by mining trouble tickets, in: Proceedings of the 20th International Conference on Management of Data, CSI, 2014, pp. 76–86.

[13] Temprado, Y., Molinero, F. J., García, C. and Gómez, J. 2008. Knowledge Discovery from Trouble Ticketing Reports in a Large Telecommunication Company, 2008 International Conference on Computational Intelligence for Modelling Control and Automation, CIMCA 2008, 37– 42.

[14] B.Bhaskara Rao, ND, a Detection Of Unfairness Online Reviews Using Deep Learning, Journal of Theoretical and Applied Information Technology, 2022, pp 4738-4778.

[15] ND, Processing Real World Datasets using Big Data Hadoop Tools, JSIR Vol.79(07), July 2020, pp 631-635.

[16] Symonenko, S., Rowe, S. and Liddy, E. D. 2006. Illuminating Trouble Tickets with Sublanguage Theory, Proceedings of the Human L

[17] Medem, A., Akodjenou, M. I. and Teixeira, R. 2009. Trouble Miner: Mining Network Trouble Tickets, 2009 IFIP/IEEE International Symposium on Integrated Network Management- Workshops, 113–119.

[18] Sample, K.R., Lin, A.C., Borghetti, B.J. and Peterson, G.L.,(2018), "Predicting Trouble Ticket Resolution", In The Thirty-First International Flairs Conference.

[19] Löfgren, J., 2017. Data Mining of Trouble Tickets for Automatic Action Recommendation.

[20] Udayaraju P, Jeyanthi P. "Early Diagnosis of Age-Related Macular Degeneration (ARMD) Using Deep Learning", Smart Innovation, Systems, and Technologies, Vol. 289, pp. 657–663, 2022.