Knowledge-based Intelligent System for IT Incident DevOps

Knowledge-based Intelligent System for IT Incident DevOps

Salman Ahmed*, Muskaan Singh*, Brendan Doherty †, Effirul Ramlan ‡
Kathryn Harkin †, Magda Bucholc*, Damien Coyle * §
*Intelligent Systems Research Centre, Ulster University, Northland Rd, Londonderry BT48 7JL, UK
†Data and Intelligent Systems, Allstate NI, Belfast BT1 3PH, UK
‡School of Computer Science, University of Galway, University Road, H91 TK33 Galway, Ireland
§Bath Institute for the Augmented Human, University of Bath, Bath BA2 7AY, UK

***Abstract*—The automation of IT incident management (i.e., handling of any unusual events that hamper the quality of IT services) is a main focus in Artificial Intelligence for IT Operations (AIOPS). The success and reputation of large-scale firms depend on their customer service and helpdesk system. These systems tend to handle client requests and track customer service agent interactions. In this research, we present a complete knowledge-based system that automates two core components of IT incident service management (ITSM): (1) Ticket Assignment Group(TAG) and (2) Incident Resolution (IR). Our proposed system bypasses the 4 core steps of the traditional ITSM process, including data investigation, event correlation, situation room collaboration, and probable root cause. It provides immediate solutions that can save companies key performance indicator(KPIs) resources and reduce the mean time to resolution (MTTR). The experiment used an industrial, real-time ITSM dataset from a prominent IT organization comprising 500,000 real-time incident descriptions with encoded labels. Furthermore, our systems are then evaluated with an open-source dataset. Compared to the existing benchmark methodologies, there is a 5% improvement in terms of Accuracy score. The study demonstrates AI automation capabilities in incident handling (TAG and IR) for large real-world IT systems.**

***Index Terms*—IT Incidents, Risk prediction, Dataset Imbalance, IT Service Management (ITSM), Information Technology Infrastructure Library (ITIL), Artificial Intelligence for IT Operations (AIOPS), Text Resolution, Assignment Group**

I. Introduction

Artificial Intelligence for IT Operations (AIOPS) aims to automate IT operations using the advances of Machine Learning (ML) and, to a certain extent Deep Learning (DL). Information Technology Service Management (ITSM) is a subset of Automatic Information Operations focusing on planning, managing, and enhancing client IT services. Due to the massive number of incident reports (IR), most IT service management businesses struggle to optimize and use their resources to prioritize and address the most important incident, leading to excessive system downtime [1]. Typically, IT workers deal directly with customers to fix difficulties with particular elements of the system and their related procedures. IT incident management is the most critical aspect of IT service management [2], [3]. The goal of the IT help desk is to register user inquiries and provide instant feedback to address those inquiries. The most common way to find these answers is to search through the solution database. IT teams must react quickly to customer and employee inquiries by notifying the appropriate departments of the escalation of the problem. High serviceability would be the main goal, and this is possible only by the rapid solution of the problem or a complete system restoration [4], [5]. Incident management starts once an incident ticket is raised. These tickets generally come from the organization (e.g., issues commonly associated with system accessibility) or the system components (i.e., where specific segments of the system issue an alert) [6], [7]. Tickets are then rated as major or minor before they are escalated to the subject matter expert. Tickets will close once the issue is fixed [8]. A subject matter expert will manually assess the problem's severity and determines whether additional investigation is required. Businesses frequently depend on manual IT ticket assignments, which regrettably leaves room for human mistakes (i.e., inaccurate level assignments) [4], [5]. Furthermore, many large organizations experience higher resource consumption from longer working hours to handle disruptions due to these human errors. Ultimately, these result in negative customer/employee feedback, directly impacting the organization's reputation [6]. For incident resolution (IR), IT service management efficiently identifies the correct solution for an incident/outage. Finding the IR against an outage is a tedious, error-prone, and painstakingly time-consuming process [8]. The manual identification of solutions for IT outages extends the Mean time to Resolution (MTTR). We plan to resolve this issue by automating the IT resource management process [8]. To do that, we have implemented the state-of-the-art DL algorithm (e.g., the Bidirectional Encoder Representations from Transformers(BERT) transformer model) to predict the solutions associated with each outage more precisely. Additionally, the assignment of IT outage tickets to an irrelevant group can cause deadlock, leading to a Major Incident Record (MIR) [9]. Similarly, this can be addressed by automating the assignment of complicated occurrences using the forecasting model of BERT to predict the Assignment group associated with each outage autonomously.

II. Related Work

One of the earliest applications of machine learning to automate the incident reporting process was carried out by Erdal [10], focusing on incident Configuration Items (CI).

The study uses the open-source machine learning application WEKA [4] to analyze data from various healthcare systems in multiple countries. Statistical-based ML approaches were considered, including Multi-nominal Naive Bayes (MNB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Decision Tree (DT). They reported that SVM attained the highest accuracy (81%) regarding the multi-label text classification of IT events for CI. On the Natural Language Processing (NLP) front (which is essential in the feature representation of text, which is ubiquitous in the majority of the incident report-based datasets), Revina et al. [11] investigate the efficacy of term frequency-inverse document frequency(TF-IDF) and linguistically-based text features in extracting the essential aspects of IT incident reports. The study was conducted using text data from the IT change management of a well-known organization. The classifiers utilized are KNN, DT, NB, Logistic Regression (LR), and SVM. Their solution is called QuickSuccess, a semisupervised classification approach that follows their earlier rule-based implementations, [12]–[15]. They analyzed classification using multiple labels (High, Medium, and Low). The tool obtained an average accuracy of 75% compared to TF-IDF, a significant gain when considering the linguistic attributes.

Zhou et al. [16] investigate a more complex scenario, including DL implementation to resolve an event. Their study is centered on two difficulties: (1) Assessing the quality of ticket resolution and (2) Offering a graded automated ITSM system based on the dataset's description and resolution reports. Before employing a regression model to quantify ticket resolution scores, they rated each ticket to improve the accuracy of the quality measurement of the tickets. They presented a technique for the automation of ticket resolution that combines a CNN model for training with the IT incident quality score produced via the quantification approach. With a precision of 74.2%, their recommended model outperformed common techniques for multi-label classification (such as neural network ensembles and hierarchical networks), suggesting that more complex models are preferable for tackling challenges in IT incident prediction/classification.

AI-based approaches to automating manual ITSM evolve with the changing technology landscape [17]. ITSM evolves to meet business-changing needs as cloud computing, artificial intelligence, and other technologies are adopted. ITSM consider an enabler of digital transformation rather than a support role, making it more customer-centric, where all recorded tickets sorts by subdomain before being sent to the appropriate department for further processing [17]. Every ticket has an incident log that can be associated with the potential to escalate into a major impacting incident. At the early stage of performing diagnoses of the incident report, IT experts manually filter through many tickets daily to determine the problem's origin. Daily ticket resolution efficiency improves by automating the IR process. Wu et al. [18] use Phrase2Vec, which enhances the text representation by mining and embedding phrases. Phrase2Vec and parsing allow words to be embedded, allowing the quantified text-mining approach to search for new patterns.

Table I
DATA DICTIONARY

Column ID	Description	Values
Public dataset (Open source benchmark data) [21]		
Short description	Brief information about the incident.	
Description	Detailed information about the incident.	
Ticket Assignment group	The group to which the incident has been assigned.	grp0,grp1
Industry-based dataset (real-time data)		
Incident number	The unique internal code of the incident	INC123xxxx
Ticket Assignment group	The group to which the incident has been assigned.	grp0,grp1
Opened at	Date/Timestamp of when created the incident record.	17/3/2020
Closed at	Date/Timestamp of when the incident record was closed.	18/3/2020
Text Resolution	The resolution solution to which incident has been reported.	
Incident severity	The level of impact for each incident.	(1 – High; 2 – Medium; 3 – Low; 4 – None)
CMDB	The name of the configuration management database associated with the incident	
Category	The category associated with the incident	
Short description	Brief information about the incident.	
Description	Detailed information about the incident.	
Status	The manual mapping from problem to incident.	(0 - MIR; 1 - Non-MIR).

Zhou et al. [19] proposed the use k-nearest neighbor approach to monitoring the resolution of tickets proposing algorithms that learn from the ticket's attributes and statistically determine the optimal solution. Muni et al. [20] proposed that DL can assist in identifying the optimal ticket for text resolution. TF-IDF vector features re-adopted, such as to reduce the number of dimensions. To produce training sets, cosine similarity is applied. Identical and separate tickets are produced using category, subcategory, and attribute-level information. The emphasis is on the issues related to ITSM systems, such as dealing with a considerable volume of service tickets daily and the potential benefits of adopting an automated system to recommend solutions to these tickets. The suggested technique identifies ticket descriptions and solutions using a DNN model. The DNN model evaluates utilizing a range of performance indicators after being trained on a dataset of Information Technology Infrastructure Library(ITIL) service tickets and measuring performance using Precision, recall, F1 score, and accuracy. The DNN-based solution outperformed rule-based and machine-learning-based methods. The authors conclude that their strategy reduces the ITSM team workload and provides accurate and rapid service ticket resolution recommendations [20].

III. Proposed System

In this study, we propose a complete knowledge-based system to mitigate the outage of IT incidents. The system's workflow is presented in Fig. 1. Like the manual process; the system escalates an incident initially handled by the service desk. Our system automates this process by predicting the relevant Ticket Assignment Group (TAG) for outages and generates a possible ticket solution. In the typical workflow, IT Assignment groups comprise IT experts that provide solutions to outages faced by the organizations. A service desk can divert the incident to the predicted TAG for expert opinion and final approval. Finally, the IT incident knowledge group analyzes the predicted solution and releases the ticket.

In contrast to the typical workflow, our proposed system bypasses the 4 essential steps, including data investigation, event correlation, situation room collaboration, and probable root cause. The system provides instant solutions by providing three core steps: observe, Engage, and Act that save companies KPI resources and mean time to resolution (MTTR). In observation, the service desk is responsible for handling IT incident outages by checking the solution dictionary & predicted TAG and forwarding it to the next stage engage. The engagement steps include the cluster of expert knowledge teams responsible for verifying and validating predicted solutions against IT outages. Agreeing to the solution will lead to the next phase, called Act; otherwise, additional expert tacit knowledge will be added against each outage to provide an exact solution to customers and forward it to Act. The Act is the final phase which is also handled by IT teams responsible for Resolving the incident and closing the Incident Log. It also ensures that the tacit knowledge should be added to the solution dictionary to update the knowledge base.

IV. Experimental Setup

Two datasets are examined in this study; (1) An industry-provided dataset involving a large IT infrastructure and (2) the public dataset available via open access [21]. The real-time dataset consists of 500,000 occurrences from a reputable IT company collected from various stakeholders (such as agency, employee, and customers). Table I listed some sample entries available in the dataset. The industrial dataset consists of real-world incident reports depicting the company's daily business activities. All transactions between January 2020 and March 2021 are tracked. It contains the following columns; `Incident number`, `Assignment group`, `Opened at`, `Closed at`, `severity`, `CMDB`, `Category`, `Short description`, `Description` and `Status`. For feature extraction, we have combined the `Short_Description` and `Description` columns into a single text column. We have performed encoding for the `Assignment_group` label, resulting in labels 0 to 38. Similar encoding schema was performed for `Resolution_text` column, resulting in labels 0 to 36. So we aim to predict an assignment group and a resolution based on the incident description.

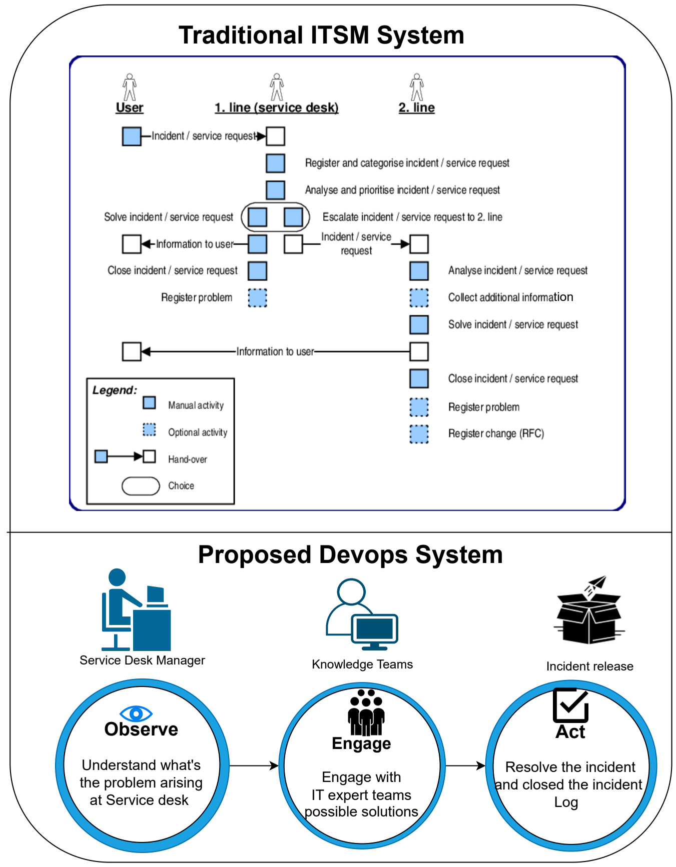

Figure 1. Comparison between the traditional ITSM workflow and the proposed DevOps solution. The proposed automation streamline the workflow by reducing the burden of manual processes and excessive inter-departmental communications.

To further validate the performance of our framework, we selected an open-source IT incident dataset [22] (as *Public* dataset). It contains the following columns; `Short_Description`, `Description`, `Caller` and `Assignment_group`. For features extraction, we have combined the `Short_Description` and `Description` columns into a single text column. We have also performed encoding for the `Assignment_group` labels, resulting in labels 0 to 49. Both datasets (*Industrial* and *Public*) represent actual day-to-day IT operations, displaying the imbalanced characteristic of IT incident labels that are normally observed in the industry.

A. Data handling, Preprocessing, and Resampling

Figure 3 illustrate the complete workflow of our proposed solution. Preliminary processing are conducted on both datasets using the natural language toolkit (NLTK) [23]. After the removal of noisy entities such as HTML Tags, stop words, punctuations, whitespace, and URLs; we normalized the data by lemmatizing, stemming, and segmenting sentences. This was followed by URL removal. The pre-processed data is then split into two, 80% for training and 20% for testing [24]. The max length parameter is set to 35 as 99% of the data lies within this length. Post-padding is performed to guarantee that the dimensions of our training and our testing datasets are consistent.

Resampling are performed using sklearn resample [25]. The resampling process involves taking new samples from the same original data pool. As a non-parametric approach to statistical inference, resampling is becoming increasingly popular [26]. It works by estimating an estimator's variability or performing statistical inference without relying on theoretical assumptions. We have adapted the sklearn Bootstrapping resampled method that involves repeatedly sampling with replacement from the original data to estimate the sampling distribution of a statistic or to calculate confidence intervals for a parameter estimate. In the default mode, a single stage of the bootstrapping operation is executed. The sklearn resample function does not just add more data points to the datasets; it also produces a random dataset resampling. Such an approach will remove the uneven data distribution, ensuring a non-bias analysis. Without resampling, the model's training often favors the labels with the largest distribution in the dataset.

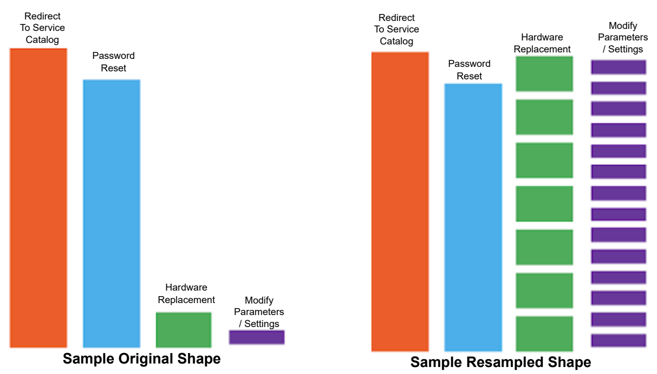

Figure 2. Resampling of datasets. The original distribution (left), and the result of the resampled dataset (right). The distribution of data is more uniform after resampling, reducing the chance of bias during training

B. Proposed algorithm

The Bidirectional Encoder Representations from the Transformers [27] or BERT employs many encoder transformers and pre-trained models. As the name implies, BERT can learn from a string of words in either the left-to-right or right-to-left direction, making it a genuinely bidirectional learning tool. Each encoder has two sublayers: the self-attention layer and the feed-forward layer. A learned BERT architecture was used with 12 encoder layers, 12 attention heads, 768 hidden size parameters, and 110 million trainable parameters. It is then trained on the event prediction issue after pre-training on 800 million unlabeled data from *BooksCorpus* and 2,500 million words from *Wikipedia*.

We performed another pre-processing step on our datasets utilising the BERTtokenizer. The classification token CLS and the sequence-entry point token SEP are selected here. This steps reshapes and tokenizes the token sequence (appended to the end of the sequence). We used the padding option PAD to fill the additional space as our generated token for representing an event was fewer than 512 tokens. Our BERT model, which has 340 million trainable parameters, generates an embedding vector with a value of 768 for each token. We ran GridSearchCV [28] using a 5-fold CV on the training dataset to hyper-parameter-tune several algorithms for computational performance. The following parameters were tuned: vocabulary size, max features, embedding dimensions, batch size, filters, kernel size, activation, loss, optimizer, and learning value. We introduced the activation function and learning rate parameters to eliminate bias and assure data linearity for DL. The parameter configurations are listed in Tab. II.

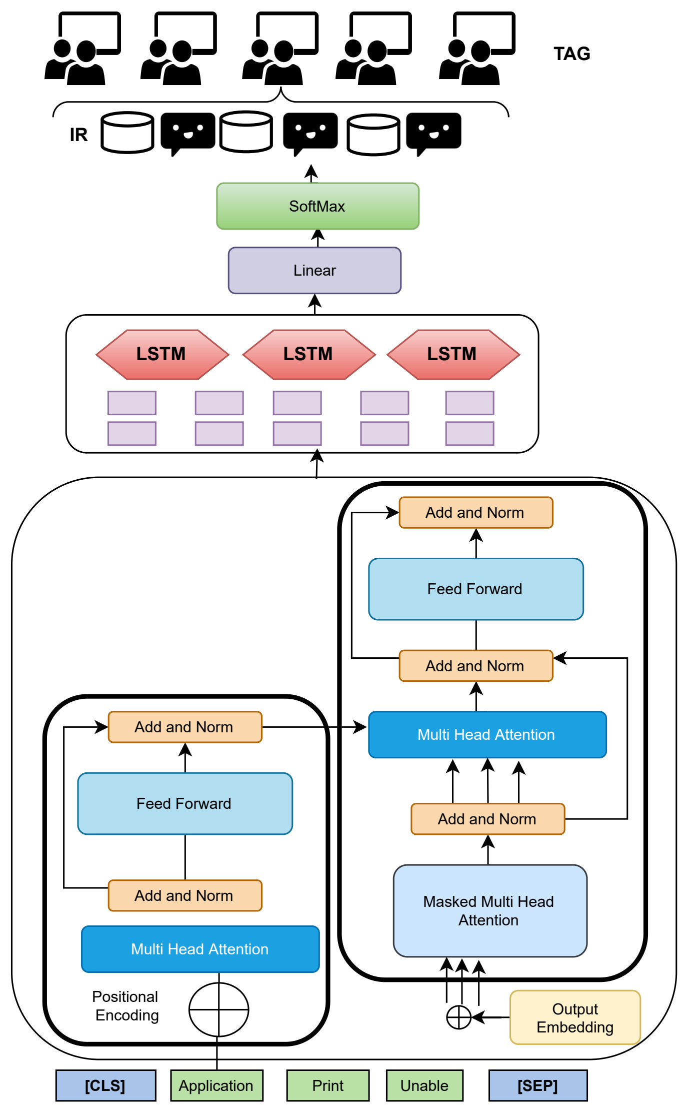

Figure 3. The proposed framework for DevOps implementation.

The output layer of the BERT model modifies for multi-class classification by adding a Long short-term memory(LSTM) layer before a fully connected layer with a softmax activation function. This layer inputs contextualized embeddings and outputs a probability distribution over the pre-defined classes. The class with the highest probability select as the predicted class for the input text. The Algorithm 1 shows the used ITSM performance metrics for our proposed system. We used the BERT Transformers model with accuracy, precision, recall, f1 score, and AUC for evaluation. As a result, we can see and assess the possible benefits of gradually implementing sophisticated algorithms [29]. To compare our model, we have selected accuracy, which is mainly reported as a benchmark performance metric. In our datasets, a higher accuracy value indicates better TAG allocation. All models are trained for 20 epochs with sets of optimum parameters. The implemented codes are available at GitHub [30], [31]

In conclusion, we have integrated the LSTM classifier with our BERT model. LSTM networks are well-suited for categorizing, interpreting, and generating predictions, as there might be delays of unknown length between significant occurrences in a data series for resampled datasets. Our approach Transformer Enhanced BERT contains the following steps:

1) First, train BERT on ITSM tokenizes data on a labeled dataset.
2) Adding a classification layer on top of BERT and training the model end-to-end.
3) BERT is fine-tuned; we use it as a feature extractor.
4) We pass the input text through the BERT model and extract the output of the last BERT layer for each token in the input text.
5) The output is a contextualized embedding that represents the meaning of the token in the context of the sentence.
6) Feed the extracted embeddings into an LSTM classifier.
7) Finally, the LSTM model takes in the sequence of embeddings and learns to classify the input text based on the task at hand.

Algorithm 1 ITSM Evaluation for Proposed system

0: $BERTModel \leftarrow LoadBERTModel()$
0: $Accuracy \leftarrow accuracy\_score()$
0: $Precision \leftarrow Precision\_score()$
0: $Recall \leftarrow Recall\_score()$
0: $f1score \leftarrow F1\_score()$
0: $AUC \leftarrow AUC\_score()$
0: **for** $Every\_IT\_incident$ **do**
0: $\quad Predicted\_BERT \leftarrow Predict(BERTModel, IT\_ticket)$
0: $\quad accuracy[] \leftarrow Accuracy(Actual\_value, Predicted\_BERT)$
0: $\quad precision[] \leftarrow Precision(Actual\_value, Predicted\_BERT)$
0: $\quad recall[] \leftarrow Recall(Actual\_value, Predicted\_BERT)$
0: $\quad f1\_score[] \leftarrow f1score(Actual\_value, Predicted\_BERT)$
0: $\quad auc[] \leftarrow AUC(Actual\_value, Predicted\_BERT)$
0: $\quad$ **End** for

V. Results and Discussions

For the *Industry* dataset, the BERT transformer model does a satisfactory job of assigning IT ticket outages to the suitable TAGs. During the initial investigation of the original (unsampled) dataset, the BERT transformer model shows a result with 82% AUC, 72% accuracy, 66% precision, 72% recall, and 67% f1-score for the assigning task. Using the same approach as TAG for the IR original dataset, we have 72% AUC, 74% accuracy, 71% precision, 74% recall, and 71% f1-score.

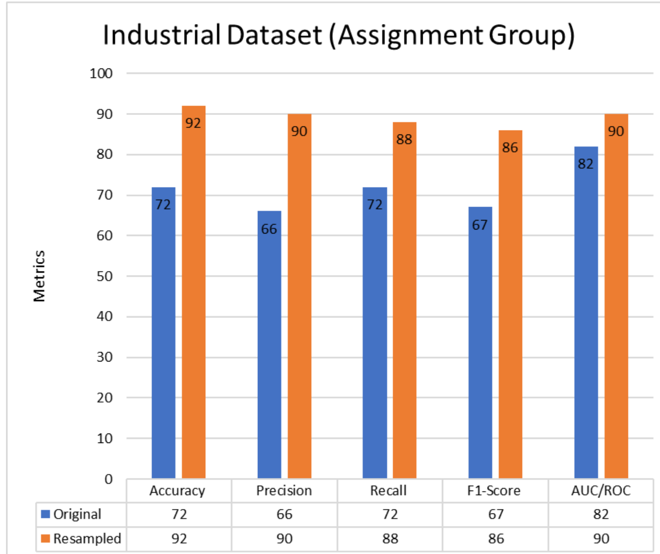

Figure 4. Industrial led dataset Assignment Group

The metrics scores are comparatively low for our BERT model. We have identified that the BERT model failed to accurately learn for multi-label classification due to the highly imbalanced data distribution [32]. To resolve this issue, we have integrated the LSTM classifier together with our BERT model. For Industrial Dataset, the results are significantly better for TAG with the resampled datasets with Transformer Enhanced BERT, with an accuracy of 92%, precision of 90%, recall of 88%, f1-score of 86%, and AUC score of 90% (figure 4). Similarly, for IR, the result is better with an accuracy of 93%, precision of 90%, recall of 91%, f1-score of 86%, and AUC of 90% (figure 5). BERT with LSTM provides masked language modeling (MLM), and as evident here enhances the prediction accuracy and allows BERT to forecast random

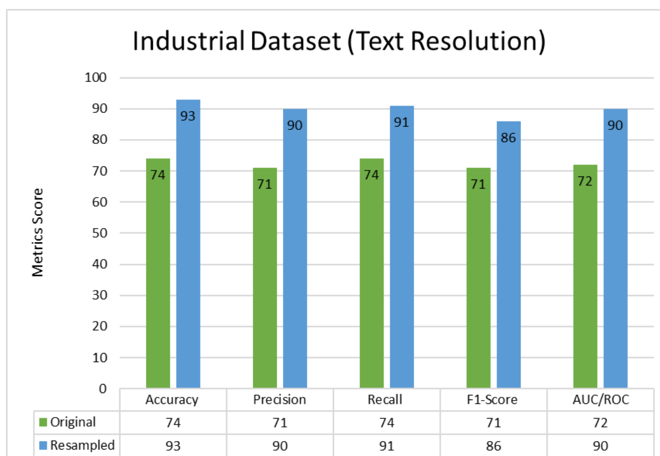

Figure 5. Public Dataset for Text Resolution

Table II
SELECTED PARAMETERS

Classifier	Vocab _size	Max _features	Embedding _dim	Batch _size	Filters	kernel	Activation	loss	optimizer	Learning _rate
	(500, 1k, 10k, 20k, 30k, 50k)	(5k, 10k, 50k, 200k)	(64, 128, 256, 512)	(8, 16, 64, 128, 256)	(200, 400, 600, 800)	(1, 2, 3, 4, None)	(Relu, Sigmoid, Gelu, Tanh)	(categorical crossentropy, crossentropy loss)	(Adam, AdamW)	(0.01, 1e-3, 1e-5)
Bert	30522	3072	512	8	768	None	Gelu	cross entropy loss	AdamW	1e-5

sample tokens in multi-class labeling during the pre-training phase.

Comparison were also made with the most recent available benchmarks in [22]. Our proposed architecture achieved 96% AUC for *BERT* compared to the benchmark Accuracy score of 57% (for *Random Forest*) , 55% (for *SVM*) , 91% (for *LSTM*), 91% (for *GRU*) and 88% (for *RNN*) reported in the original study figure 6). These results demonstrate that transformer-based models can deal with the non-standard, non-conforming representations of real-world incident reports, including their varying lengths and vocabularies better than the other ML approaches.

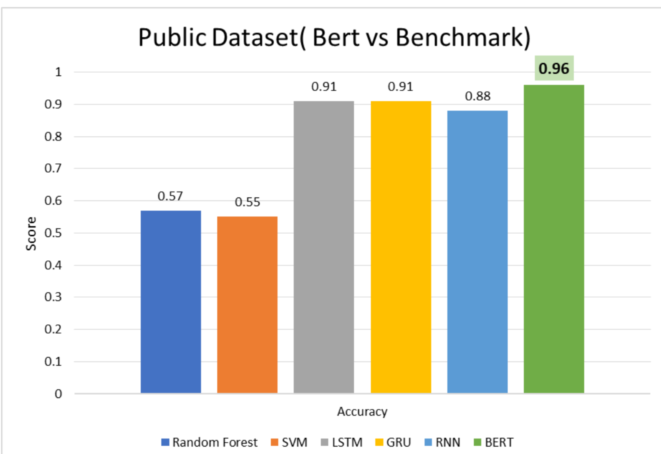

Figure 6. Validated results using Public datasets

VI. Conclusion

In this study, we have developed a solution to autonomously manage IT incident reports/ticket for IR and TAG. Our approach has the potential to provide an instant resolution to incoming IT incident tickets. Additionally, we also proposed a novel framework to automate the IT incidents for TAG allocation. We conducted a series of experiments using two different datasets to demonstrate that advanced Transformer models, like Transformer Enhanced BERT, can handle the unbalanced features typically associated with IT incident report databases. To evaluate further its capabilities as automated ITSM system in large-scale IT companies, an evaluation of the proposed pipeline in a real-time operational scenario will be carried out.

VII. ACKNOWLEDGMENT

We are grateful for access to the Tier 2 High-Performance Computing resources provided by the Northern Ireland High-Performance Computing (NI-HPC) facility funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant Nos. EP/T022175/ and EP/W03204X/1. Damien Coyle is supported by the UKRI Turing AI Fellowship 2021-2025 funded by the EPSRC (grant number EP/V025724/1). Salman Ahmed is supported by a Dr. George Moore Ph.D. scholarship.

References

[1] "The cost of downtime," https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/, 2020, accessed: 2022-04-14.

[2] Ž. Deljac, M. Randić, and G. Krčelić, "Early detection of network element outages based on customer trouble calls," *Decision Support Systems*, vol. 73, pp. 57–73, 2015.

[3] S. Ahmed, M. Singh, B. Doherty, E. Ramlan, K. Harkin, M. Bucholc, and D. Coyle, "An empirical analysis of state-of-art classification models in an it incident severity prediction framework," *Applied Sciences*, vol. 13, no. 6, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/6/3843

[4] A. Revina, K. Buza, and V. G. Meister, "It ticket classification: The simpler, the better," *IEEE Access*, vol. 8, pp. 193 380–193 395, 2020.

[5] J. Costa, R. Pereira, and R. Ribeiro, "Itsm automation-using machine learning to predict incident resolution category," *ITSM automation-Using machine learning to predict incident resolution category*, pp. 5819–5830, 2019.

[6] C. Bartolini, M. Sallé, and D. Trastour, "It service management driven by business objectives an application to incident management," in *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*. IEEE, 2006, pp. 45–55.

[7] S. Ahmed, M. Singh, B. Doherty, E. Ramlan, K. Harkin, M. Bucholc, and D. Coyle, "Ai for information technology operation (aiops): A review of it incident risk prediction," in *International Conference on Soft Computing Machine Intelligence*. United States: IEEE, Jul. 2022.

[8] W. Zhou, T. Li, L. Shwartz, and G. Y. Grabarnik, "Recommending ticket resolution using feature adaptation," in *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE, 2015, pp. 15–21.

[9] S. Ahmed, M. Singh, B. Doherty, E. Ramlan, K. Harkin, M. Bucholc, and D. Coyle, "An empirical analysis of state-of-art classification models in an it incident severity prediction framework," *Applied Sciences*, vol. 13, no. 6, pp. 1–27, Mar. 2023.

[10] E. S. İşcen and M. Z. Gürbüz, "A comparison of text classifiers on it incidents using weka," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019, pp. 510–515.

[11] A. Revina, K. Buza, and V. G. Meister, "Designing explainable text classification pipelines: Insights from it ticket complexity prediction case study," *Interpretable Artificial Intelligence: A Perspective of Granular Computing*, vol. 937, p. 293, 2021.

[12] N. Rizun and A. Revina, "Business sentiment analysis. concept and method for perceived anticipated effort identification," in *Proceedings of the 28th International Conference on Information Systems Development (ISD2019). At: Toulon, France*, 2019.

[13] N. Rizun, A. Revina, and V. Meister, "Method of decision-making logic discovery in the business process textual data," in *International Conference on Business Information Systems*. Springer, 2019, pp. 70–84.

[14] N. Rizun, V. Meister, and A. Revina, "Discovery of stylistic patterns in business process textual descriptions: It ticket case," *Innovation Management and Education Excellence through Vision*, 2020.

[15] S. S. Ali Zaidi, M. M. Fraz, M. Shahzad, and S. Khan, "A multiapproach generalized framework for automated solution suggestion of support tickets," *International Journal of Intelligent Systems*, 2021.

[16] W. Zhou, W. Xue, R. Baral, Q. Wang, C. Zeng, T. Li, J. Xu, Z. Liu, L. Shwartz, and G. Ya. Grabarnik, "Star: A system for ticket analysis and resolution," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 2181–2190.

[17] J. Xu, J. Mu, and G. Chen, "A multi-view similarity measure framework for trouble ticket mining," *Data Knowledge Engineering*, vol. 127, p. 101800, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169023X18304580

[18] Y. Wu, S. Zhao, and W. Li, "Phrase2vec: Phrase embedding based on parsing," *Information Sciences*, vol. 517, pp. 100–127, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519311429

[19] W. Zhou, L. Tang, T. Li, L. Shwartz, and G. Y. Grabarnik, "Resolution recommendation for event tickets in service management," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 287–295.

[20] D. P. Muni, S. Roy, Y. T. Y. J. J. L. Chiang, A. J.-M. Viallet, and N. Budhiraja, "Recommending resolutions of itil services tickets using deep neural network," in *Proceedings of the Fourth ACM IKDD Conferences on Data Sciences*, ser. CODS '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3041823.3041831

[21] salmanahmed1993/it-ticket-classification.xlsx at main · salmanahmed1993/salmanahmed1993. [Online]. Available: https://github.com/salmanahmed1993/salmanahmed1993/blob/main/IT-Ticket-Classification.xlsx

[22] G. Kavitha-Kothandaraman. Kavitha-kothandaraman/automatic-it-ticket-assignment-nlp: To build an ai-based classifier model to assign the tickets to right functional groups by analyzing the given description. [Online]. Available: https://github.com/Kavitha-Kothandaraman/Automatic-IT-Ticket-Assignment-NLP

[23] S. Bird and E. Loper, "NLTK: The natural language toolkit," in *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 214–217. [Online]. Available: https://aclanthology.org/P04-3031

[24] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, "A critical look at the current train/test split in machine learning," 2021. [Online]. Available: https://arxiv.org/abs/2106.04525

[25] sklearn resample. sklearn.utils.resample — scikit-learn 1.2.0 documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

[26] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Springer, 1993. [Online]. Available: https://doi.org/10.1007/978-1-4899-4541-9

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[28] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for nas," 2019. [Online]. Available: https://arxiv.org/abs/1912.06059

[29] F. Al-Hawari and H. Barham, "A machine learning based help desk system for it service management," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 6, pp. 702–718, 2021.

[30] S. Ahmed. salmanahmed1993. [Online]. Available: https://github.com/salmanahmed1993/salmanahmed1993/blob/main/ITSM_Assignment_Transformer_bert_submitted.ipynb

[31] "salmanahmed1993resoultion." [Online]. Available: https://github.com/salmanahmed1993/salmanahmed1993/blob/main/ITSM_Severity_Transformer_bert_resolution-submitted.ipynb

[32] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works." arXiv, 2020. [Online]. Available: https://arxiv.org/abs/2002.12327