Jurnal Ilmiah Informatika dan Komputer (Informatech)
Volume 3, Number 1, June 2026, page 85- 95
DOI : 10.69533
Website : *https://ejournal.rizaniamedia.com/index.php/informatech*

E-ISSN: 3047-4752

Three-Category News Text Classification Using LSTM on a Local Government Portal

Febrisari Amalia[1*], Andhy Permadi[2], Ilham[3]

[*1,2,3] Universitas Islam Negeri Sunan Ampel Surabaya, Jl. Dr. Ir. H. Soekarno No.682, Surabaya, 60294
[*1]*email : febrisariamaliaa@gmail.com*
[2]*email: andhy@uinsa.ac.id*
[3]*email: ilham@uinsa.ac.id*

(*Article Received: 24 April 2026; Article Revised: 19 May 2026; Article Published: 1 June 2026;*)

ABSTRACT *– This research develops an automated three-category news classification system for the official news portal of the Regional Revenue Agency (Bapenda) of Surabaya City, using a Long Short-Term Memory (LSTM) deep learning model to reduce reliance on manual, inconsistent categorization. The training corpus combined original institutional data, web-scraped articles from other regional revenue agencies, and GPT-generated synthetic text, all validated by domain experts, then processed through a structured preprocessing pipeline. On an unseen test set of 513 authentic institutional articles, the model achieved an overall accuracy of 77.78% and a weighted F1-score of 0.78, driven mainly by strong performance on the majority "Important Information" class (F1-score 0.87). However, the macro-averaged F1-score was only 0.4434, reflecting substantially weaker performance on minority classes, particularly "Information Technology" (F1-score 0.069). The model was deployed as a human-in-the-loop recommendation feature within the agency's existing dashboard.*

Keywords *- Text Classification; Long Short-Term Memory; Government Portal.*

Klasifikasi Teks Berita Tiga Kategori Menggunakan LSTM pada Portal Pemerintah Daerah

ABSTRAK *– Penelitian ini mengembangkan sistem klasifikasi berita otomatis untuk tiga kategori pada portal berita resmi Badan Pendapatan Daerah (Bapenda) Kota Surabaya, menggunakan model deep learning Long Short-Term Memory (LSTM) untuk mengurangi ketergantungan pada pengategorian manual yang tidak konsisten. Korpus pelatihan menggabungkan data asli institusi, artikel hasil scraping dari portal Bapenda daerah lain, dan teks sintetis buatan GPT yang telah divalidasi oleh pakar domain, lalu diproses melalui pipeline preprocessing terstruktur. Pada data uji tak-terlihat sebanyak 513 artikel institusi asli, model mencapai akurasi keseluruhan 77,78% dan F1-score berbobot 0,78, terutama didorong oleh performa kuat pada kelas mayoritas "Info Penting" (F1-score 0,87). Namun, F1-score makro hanya 0,4434, menunjukkan performa yang jauh lebih lemah pada kelas minoritas, terutama "Teknologi Informasi" (F1-score 0,069). Model ini diterapkan sebagai fitur rekomendasi human-in-the-loop pada dashboard milik instansi.*

Kata Kunci *– Klasifikasi Teks; Long Short-Term Memory; Portal Pemerintah.*

1. INTRODUCTION

The advent of digital information and communication technologies has led to complete transformation of the operational paradigm of governmental institutions around the world. In the current era of digitalization, e-government projects are not only beneficial but crucial imperatives for improvement of public service delivery, ensuring institutional transparency and civic engagement. Web portals have thus become the primary, easily accessible and authoritative platform for disseminating information by regional and local governments. They are widely used to post about the institution's activities, communicate critical changes in policies and inform about public services operation to a wide range of demographics. The usability, accessibility and overall integrity of those digital platforms become highly important since it defines the success of the communication efforts of the government and people's ability to access required information [1], [2].

Beyond simply disseminating information, government portals also function as a long-term record-keeping resource: every item published eventually becomes part of a searchable archive that citizens, researchers, and internal staff may need to consult later. How each publication is labeled therefore matters well beyond the moment it goes live, since it directly shapes how easily that information can be found again months or even years down the line.

For the Regional Revenue Agency (Badan Pendapatan Daerah, or Bapenda) of Surabaya City specifically, the official news portal plays a broad and genuinely important role. It hosts everything from tax socialization campaigns and administrative notices to internal coordination meetings, infrastructure upgrades, and field visits. Yet early observations for this research uncovered a significant bottleneck in the agency's digital workflow: news publications are still classified and categorized entirely by hand. Each time a new article is written, the assigned administrator has to read it, interpret its meaning, and manually pick a category from a drop-down menu.

Simple as that step sounds, it happens for every article that goes through the editorial pipeline — which, in a reasonably active agency, can mean dozens of individual classification decisions in a single week. Over time, the staff hours spent on this repetitive task add up to a real opportunity cost, pulling attention away from more valuable communication work.

This manual approach also becomes harder to sustain as time goes on. As the agency publishes more digitally, sorting everything by hand simply takes longer. It's also inherently subjective, since it relies on individual judgment: different administrators, with different backgrounds and reading of the language, will often categorize the same kind of article differently. A long piece about a new tax-payment app, for instance, might be filed under "Information Technology" by one person because it's about a digital tool, while another might file it under "Important Information" because it's directly useful to taxpayers. Multiply that inconsistency across hundreds of articles, and the database starts to suffer, search results become less reliable, and users get confused trying to find what they need.

These accumulated inconsistencies chip away at the news archive's usefulness as a retrieval tool, making it progressively harder for both staff and the public to locate older material through the portal's category-based navigation.

This is essentially where Natural Language Processing (NLP), and more specifically automated text classification, comes in. In simple terms, it's the use of machines to read unstructured text and assign it labels based on meaning and word usage. As the volume of digital publications keeps growing, this kind of automation becomes less of a convenience and more of a necessity, it's what allows large collections of information to stay organized, searchable, and consistent [3].

Automated text classification itself has gone through several distinct phases over the years: it started with simple rule-based keyword matching, moved on to statistical machine learning models built on hand-crafted features, and has more recently arrived at deep learning architectures that learn representations directly from raw text. Each phase has generally required less manual engineering while getting better at generalizing to text it hasn't seen before.

Over the last decade, deep learning techniques have brought about the complete rethinking of the whole paradigm of text classification, outperforming classical approaches such as Support Vector Machines (SVM), Naive Bayes and Random Forests in extraction of highly dimensional representations of complex texts [4]. Among many deep learning neural network architectures, the Long Short-Term Memory (LSTM) model proved to be extremely robust and efficient for NLP tasks. LSTM is a highly specialized form of Recurrent Neural Network (RNN) designed for sequential data processing. Contrary to regular feed-forward networks, LSTM has a highly complex inner memory state that allows it to capture interrelations within the long sequences of words. This technique makes the model perfect for analysis of highly contextual and quite lengthy paragraph structures that are characteristic to governmental news publications [5], [6].

While newer Transformer-based architectures such as BERT and its Indonesian variant, IndoBERT, have achieved strong benchmark results across many text classification tasks, this research deliberately selected LSTM for three concrete, dataset-specific reasons. First, prior comparative studies indicate that LSTM-based architectures remain competitive with, and in some small-data settings even preferable to, pretrained Transformer models, where the Transformer's much larger parameter count increases the risk of overfitting rather than delivering a clear performance benefit [7]. Second, the labeled training corpus available in this study, after preprocessing, produced a vocabulary of only 534 unique tokens, a scale far below what Transformer-based models are typically pretrained and fine-tuned to exploit effectively. Third, and most decisively, the model in this research was trained and is intended to run entirely on a CPU-only computing environment, reflecting the actual infrastructure available to a regional government revenue agency; the substantially higher computational and memory

demands of fine-tuning and serving Transformer-based models make them impractical under this constraint, whereas LSTM offers an efficient, low-latency alternative suited to the agency's existing production environment [8].

That said, even though LSTM networks have repeatedly proven effective for text classification tasks like movie review sentiment analysis or basic spam detection, applying them to a more specific and less common domain like a local government portal brings its own challenges. The most immediate one is a shortage of labeled data. Deep learning models are notoriously data-hungry, they typically need thousands of labeled examples to learn and perform reliably. To work around this in the present research, the lack of data was addressed through advanced web scraping targeting related organizations, combined with synthetic data generated using modern Large Language Models. Every piece of data gathered this way was then carefully reviewed by people with subject-matter expertise [9].

This combined strategy, drawing on external, structurally similar government text alongside carefully curated synthetic samples, reflects a broader trend in applied NLP research, where practitioners increasingly turn to creative data-engineering solutions to make up for the scarcity of domain-specific labeled corpora in specialized or under-resourced institutional settings.

The main goal of this research is to build a system that can automatically suggest news categories, using an LSTM-based deep learning architecture. It's being developed specifically for the Bapenda Surabaya news portal, which needs help sorting news into three categories: Important Information, Information Technology, and General. The plan is to integrate this system directly into the ASP.NET Core MVC dashboard the agency already uses, so that it can predict the appropriate category in real time — as soon as someone enters a piece of text, the system immediately suggests a category. This is meant to make publishing news faster and easier, reduce classification errors, and keep everything better organized. In short, the recommendation system is designed to make the Bapenda Surabaya team's job simpler and more efficient.

In doing so, this research also contributes a practical case study of applying deep learning–based text classification within an Indonesian regional government context, an area still comparatively underexplored in the existing literature compared to more widely studied domains such as commercial product reviews or international news outlets.

2. LITERATURE REVIEW

In general, text classification is considered one of the most basic and widely studied problems in the field of Natural Language Processing (NLP). Its primary task is the automatic organization, structuring, and routing of huge amounts of unstructured textual data into discrete semantic classes. In the current government, corporate, and organizational environment, this technology is used universally for various critical processes, including document routing, real-time public sentiment analysis, spam and phishing detection, and categorization of archival news and public relations broadcasting. The basic computational process consists of transforming unstructured textual data into high-dimensional mathematical representations, thus allowing for machine learning algorithms to discover hidden statistical patterns and correctly classify them into specific classes [3], [4].

Beyond pure text classification, the broader field of NLP also covers related problems like sentiment analysis, topic modeling, and named entity recognition. All of these ultimately rely on the same basic requirement: turning unstructured language into structured, machine-readable representations before any further analysis can happen.

How well a text classification model performs in the end depends heavily on the quality of its raw input data and how rigorously that data is preprocessed. Raw human language tends to be messy, full of grammatical noise, syntactic variation, typos, and complex morphology. Preprocessing exists precisely to deal with this: it strips out the noisy elements, standardizes vocabulary, and helps manage the sheer dimensionality that comes with natural language. A good deal of academic research backs this up, showing that a systematic, multi-level preprocessing pipeline, covering thorough punctuation cleaning, case folding, tokenization, removal of relevant stopwords, and morphological normalization such as stemming or lemmatization — genuinely improves a neural model's ability to recognize topical patterns and cut down on unnecessary variance [10], [11].

In practical terms, each preprocessing stage addresses a distinct source of textual noise: cleaning removes non-linguistic artifacts such as URLs and stray punctuation, case folding prevents the model from treating capitalized and lowercase forms of the same word as different tokens, and stopword removal eliminates high-frequency function words that carry little discriminative meaning for classification purposes.

Classically, text classification has been widely based on the use of traditional statistical machine learning algorithms that needed extensive manual feature engineering from the part of data scientists, such as generating complex TF-IDF matrices or N-grams. Nowadays, deep learning techniques have completely changed the situation as they allow

learning of hierarchical dense representations directly from the raw sequential data. Recurrent Neural Networks (RNN) have gained tremendous popularity among the text processing community due to their sequential processing properties similar to the way humans process textual data, reading it word-by-word. However, standard RNNs face the serious problem of vanishing gradient during backpropagation, thus being unable to learn and retain long-term dependencies in lengthy texts, like news articles [5].

The vanishing gradient problem essentially means that, as information is propagated backward through many time steps during training, the gradient signal used to update earlier weights becomes progressively smaller, until it is no longer strong enough to meaningfully influence the learning process, effectively causing the network to forget information encountered early in a long sequence.

To address these inherent limitations of standard RNNs, the Long Short-Term Memory (LSTM) architecture was developed. LSTM works through an internal memory cell governed by three gates: the input gate, forget gate, and output gate. The forget gate decides how much of the previous memory state to discard; the input gate determines how much new information gets added to that memory state; and the output gate controls how much of the current internal memory gets passed on to the next layer. Together, these gates dynamically manage the flow of information, allowing the network to hold onto relevant context from earlier in a document while filtering out syntactic details that aren't needed. This is what lets the model preserve the overall meaning of an entire paragraph or article by the time it reaches the final classification step [6].

In effect, the three gates work together to give the network a form of selective memory — retaining meaningful context, like the main subject introduced in a news article's opening sentence, while gradually letting go of grammatical filler that doesn't really contribute to the classification decision.

Comparative studies on Indonesian-language text classification reinforce this architectural choice. Koto et al. introduced IndoBERT as a Transformer-based benchmark specifically pretrained for Indonesian NLP tasks [8], and subsequent work applying IndoBERT to Indonesian news classification has shown strong results on large, well-balanced corpora [12]. However, more recent comparative work evaluating LSTM, attention-augmented BiLSTM, and fine-tuned IndoBERT side-by-side on an imbalanced Indonesian news classification task found that Transformer-based models do not automatically outperform recurrent architectures once class imbalance and limited data availability are taken into account [13]. This suggests that the choice between LSTM and Transformer-based models should be driven by the specific characteristics of the dataset and deployment environment, rather than by architectural novelty alone — a consideration directly relevant to the present study's small, imbalanced, and institution-specific corpus.

One of the major and significantly difficult impediments for applying cutting-edge deep learning models in niche and unique domains (for example, the government revenue portal of a certain region) is the critical shortage of properly labeled training data specific to the domain in question. Deep Neural Networks, by nature of the architecture, are data-heavy models that require at least thousands and even tens of thousands of annotated samples to generalize without overfitting. To effectively tackle this data scarcity issue, modern researchers use sophisticated Data Augmentation methods. While classic text augmentation techniques included such primitive operations as random synonym replacements, word deletions, and circular back translations, modern approach switched to a completely generative one [14].

Traditional augmentation methods, while computationally inexpensive, often introduce a degree of unnatural phrasing into the generated text, since operations such as random word substitution do not account for the broader grammatical or semantic context of the sentence being modified.

Increasingly, AI researchers resort to using the revolutionary capabilities of the Large Language Models (LLM) and other generative artificial intelligence platforms (such as the GPT-based architecture) for generating quality, contextually and grammatically correct text data [15], [16], [17]. The application of generative models in text generation for training sets proves to be extremely promising in terms of increasing the volume of the dataset and addressing class imbalance issues without losing semantic fidelity of the domain. However, synthetic text generation should be used extremely carefully since unsupervised or incorrectly prompted generation may result in algorithmic hallucinations, factual errors, or topical shift which will poison the whole dataset. Thus, when using synthetic data for training the classifier in specific organizational categories, thorough validation by experienced domain experts is strictly mandatory [18]. It makes sure that the synthetic samples correspond to the semantic and operational standards of the institution in question [19].

This human validation step is particularly important in a bureaucratic setting, where the connotations, formality level, and institutional terminology of a text can subtly signal which category it belongs to, nuances that a generic language model prompted without sufficient domain

context may fail to reproduce accurately.

3. RESEARCH METHODS

This research was carried out following a rigorous software engineering and data science development lifecycle. The methodology is organized into several key stages: problem identification and category definition, data acquisition and augmentation, linguistic preprocessing, deep learning model training, performance evaluation, and full-stack integration and deployment of the final solution.

Problem Identification and Category Definition

The first phase of the analysis involved a series of discussions and structured interviews with the management and IT teams at Bapenda Surabaya. The goal was to understand the current limitations of their digital publication workflow and to establish clear boundaries for the new classification model. Through this collaboration, it became clear that the portal's needs could best be served by three mutually exclusive news categories. Getting these category definitions right was essential, since they would form the basis for all subsequent data labeling and model training.

Table 1. Defined News Categories and Operational Criteria

Category	General Criteria
Important Information	High-priority information, critical institutional announcements, urgent policy changes, essential taxpayer services, or information that needs to be acted upon immediately.
Information Technology	Publications only relating to digital transformations, new digital systems, software applications, or IT infrastructure updates in the agency.
General	Routine institutional activities, simple documentation of events, coordination meetings, working visits, and routine public relations information.

Data Collection and Multi-Source Augmentation

Building a strong and accurate neural network necessarily entails having a sufficiently large, rich, and balanced set of training documents. In the first place, the initial training corpus has been taken from the historical data of the Bapenda Surabaya portal in its old SQL database form. Nevertheless, during an exploratory data analysis, it turned out that this historical training dataset is extremely small and unbalanced across three newly introduced categories. Thus, special data acquisition and augmentation methods had to be used.

First of all, automated web scraping via parsing Python libraries has been applied to scrape analogous news articles from the official public portals of other local revenue authorities of Indonesian cities and provinces (of the revenue agencies of of the revenue agencies of East Java Province, Malang, Blitar, Mojokerto, Pasuruan, Padang, and Madiun) [20], [21], [22], [23], [24], [25], [26], [27], [28]. As a result, there has been obtained a vast and highly related source of government-styled bureaucratic text. Secondly, a generative AI model based on GPT-5.5 architecture has been prompted in compliance with domain-specific restrictions to generate artificial documents. The generation has been explicitly focused on the two minority categories of Information Technology and General to achieve the balance in the dataset and avoid serious majority class bias of the LSTM model [16], [17].

Table 2. Dataset Sources, Types, and Composition Strategy

Data Source	Data Type	Description / Purpose	Approx. Sample Count
Surabaya Revenue Agency Portal	Original Data	Primary ground-truth data fitting the exact institutional context and terminology.	Merged into 9,000-sample pool (see note below)
Other Local Govt Portals	Scraped Data	Supporting data adding bureaucratic vocabulary and geographical variety.	Merged into 9,000-sample pool (see note below)
Generative AI (GPT-5.5)	Synthetic Data	Generated data prompted to balance class distributions.	Merged into 9,000-sample pool (see note below)
Final Validated Data	Validated Data	Master dataset, reviewed by domain experts, zero conceptual drift.	513 (Info Penting 431 / Umum 64 / Teknologi Informasi 18)

The Original, Scraped, and Synthetic data described above were merged and expert-validated prior to modeling into a single labeled pool of 9,000 samples, perfectly balanced across the three target categories (3,000 samples each). This pool was the exclusive source for the Training and Validation sets in Table 3. Because the merge was performed upstream of the modeling pipeline, the exact sub-totals contributed by each of the three original feeder sources were not separately retained. In contrast, the Final Validated Data (513 samples) was kept entirely separate and reflects the natural, imbalanced class distribution actually observed on the agency's portal (Info Penting 431; Umum 64; Teknologi Informasi 18);

this set was used exclusively as the primary test set and contains no synthetic or scraped samples.

In order to guarantee the conceptual soundness of the training process of the deep learning model, all gathered texts – especially the scraped and synthetic parts – have undergone a thorough manual validation stage carried out by senior PR domain experts of the agency. All texts failing to satisfy the agency's linguistic requirements and category definitions have been eliminated. Finally, the purified dataset has been randomly stratified to avoid any data leaks during the training and validation processes.

Table 3. Dataset Distribution and Stratification Strategy

Data Partition	Sample Amount	Primary Function
Training Set	5,100	Utilized exclusively to feed the neural network and adjust the LSTM mathematical weights via backpropagation.
Validation Set	900	Utilized to monitor the training process epoch-by-epoch, tune hyperparameters, and prevent model overfitting.
Testing Set	513	Kept completely isolated until the end; used purely to evaluate the final model's objective predictive performance.

Comprehensive Text Preprocessing Pipeline

In its original form, the harvested raw texts include noisy HTML tag fragments, random punctuation, non-conforming capitalization, and numerals unrelated to the meaning. The implementation of a carefully constructed and organized multi-step text preprocessing pipeline allowed the systematic cleaning and normalization of the corpus to produce text in the machine-readable, highly structured form that is specifically optimized for high-dimensional word embedding [10], [11].

Table 4. Sequence of Text Preprocessing Steps and Transformations

Preprocessing Stage	Example Input Text (Before)	Example Output Text (After)
Cleaning (Regex)	Sosialisasi pajak 2025!!! http://link.com	sosialisasi pajak
Case Folding	Sistem Informasi BAPENDA Jatim	sistem informasi bapenda jatim
Tokenizing	portal berita bapenda	[portal, berita, bapenda]
Stopword Removal	kegiatan ini dilaksanakan oleh bapenda	kegiatan dilaksanakan bapenda
Normalization	info ttg denda pjk	informasi tentang denda pajak

After the thorough linguistic normalization of the text, the resulting standardized tokens needed vectorization. A Tokenizer was fit exclusively on the vocabulary of the training corpus to construct a dictionary that mapped each word into an integer token. As the deep neural networks demand a constant dimensionality of input tensors to be able to process the data in batches, sequence padding techniques were utilized. Sequences that exceeded the predetermined threshold of length were truncated, whereas the shorter ones had been padded with zeros to produce strictly constant dimensionality tensors to be fed into the LSTM input embedding layer.

Deep Learning Model Architecture and Training

The main classification model was created using TensorFlow and Keras as the deep learning tools. The neural structure begins with a dimensional Embedding layer. The important thing about this layer is that it changes the sparse integer tokens into dense vector spaces. This helps in capturing the deep meaning connections, similarities and other details between the words, in the vocabulary. After that come the LSTM processing layers. These layers are set up to pass the word embeddings into the network step by step. They use their gating systems to understand the context and take out semantic features from the entire documents. Some Dropout layers were placed between the repeating layers to stop neuron interactions and strongly fight the overfitting issue [4].

Table 5. Neural Network Training Parameters and Architecture

Hyperparameter / Architectural Setting	Configured Value
Framework	TensorFlow 2.20 / Keras
Embedding Dimension	128
LSTM Units	128
Spatial Dropout (after Embedding layer)	0.3
LSTM Dropout / Recurrent Dropout	0.3 / 0.2
Dense Layer Units	64
Dense Dropout Rate	0.5
Actual Vocabulary Size	534 tokens (cap set at 20,000)
Maximum Sequence Length	101 tokens

Hyperparameter / Architectural Setting	Configured Value
Total Training Samples	5,100
Total Validation Samples	900
Optimizer Algorithm	Adam (Adaptive Moment Estimation), learning rate 0.001
Loss Function	Sparse Categorical Crossentropy
Output Layer Activation	Softmax
Batch Size	32
Maximum Epochs	50, with Early Stopping (patience 5, monitor val_loss, restore best weights)
Learning Rate Scheduling	ReduceLROnPlateau (factor 0.5, patience 2, min_lr 1e-6)
Target Classification Classes	3 (Multi-class Formulation)
Hardware / Environment	Google Colab, CPU-only runtime, Python 3

All architectural and training parameters listed in Table 5 correspond directly to the final implementation used to produce the results reported in this paper, and are provided in full to support independent replication of the training pipeline.

Mathematically, the network architecture ends with the Dense layer, fully connected layer with the Softmax activation function. The equation for Softmax mathematically translates the unbound logits output scores of the LSTM layers into properly normalized probability distributions making sure that the sum of the probabilities across the three target categories is exactly equal to one. The whole model was compiled using highly efficient Adam optimizer alongside with the Sparse Categorical Crossentropy loss function, perfectly suitable for mutually exclusive multi-class problems. Training was conducted using multiple epochs and stopped automatically when the validation loss reached the plateau to provide optimal generalization.

4. RESULTS

Training performed efficiently and demonstrated extremely consistent and stable learning behavior. Analyzing in detail training logs and epoch history revealed logarithmic increase of training accuracy values, which was very precisely mirrored by validation accuracy trajectories. At the same time, the values of the loss function were decreasing steadily. The complete lack of significant divergence between the training and validation curves proves empirically that the model succeeded in abstraction of the semantic patterns of language without simple memorization of training dataset.

Overall Model Predictive Evaluation

In order to conclude about the objective prediction quality of the LSTM model and check its applicability in real-world bureaucratic situation, the model was evaluated against completely unseen testing dataset, consisting of 513 manually validated and pristine articles. The evaluation was performed using standard machine learning metrics: Overall Accuracy, Precision, Recall, and the F1-score calculated with two approaches (Macro and Weighted averaging) [29]. As detailed in the dataset composition note above, this 513-article test set consists exclusively of authentic, previously unseen institutional data, with no synthetic or scraped samples included in the evaluation reported below.

Table 6. Overall Model Evaluation Metrics on Unseen Test Data

Evaluation Metric	Calculated Statistical Value
Final Test Loss	2.839
Overall Predictive Accuracy	77.78%
Macro-Averaged Precision	0.4387
Macro-Averaged Recall	0.4594
Macro-Averaged F1-score	0.4434
Weighted-Averaged Precision	0.7966
Weighted-Averaged Recall	0.7778
Weighted-Averaged F1-score	0.7851

Fully trained model demonstrated extremely high overall accuracy of 77.78%, mathematically indicating that more than three quarters of completely unseen publications were predicted flawlessly. But an important note should be made about huge mathematical difference between the values of the Macro-averaged F1-score (0.4434) and Weighted-averaged F1-score (0.7851). This large gap shows very clearly the profound impact of the class imbalance existing inside testing dataset. It implies that although neural network performs extremely well on the majority class, its performance fluctuates greatly when dealing with the minority ones.

Granular Category-Specific Performance

Even a very minute and much more detailed analysis of the evaluation metrics segregated for the particular categories helps in obtaining invaluable insights into the specific cognitive skills and natural linguistic weaknesses of the neural network.

Table 7. Granular Evaluation Metrics Segregation for Specific Category

Target Category	Precision	Recall	F1-score	Total Data Support
Important Information	0.8954	0.8538	0.8741	431

Target Category	Precision	Recall	F1-score	Total Data Support
Information Technology	0.0909	0.0556	0.0690	18
General	0.3297	0.4688	0.3871	64

It can be easily noted from the above statistics that the LSTM model shows a highly dominating performance in recognizing 'Important Information' and gives a very reliable F1-score of 0.8741. It is indeed a very good result since giving priority to important information of taxpayers is the first objective of the organization. On the other hand, the model performed very poorly while trying to identify 'Information Technology' news (giving a very poor F1-score of 0.0690) and regular 'General' administration actions (F1-score 0.3871).

Confusion Matrix Diagnostics

To determine the exact root cause of the prediction failure for the minority classes, a very detailed cross-sectional analysis of the Confusion Matrix has been made. The Confusion Matrix helps to identify the exact point of classification contamination in multidimensional vector space.

Table 8. Prediction Mapping using Confusion Matrix

Actual True Category	True Positive (Correct)	False Negative (Wrong)	Total Data
Important Information	368	63	431
Information Technology	1	17	18
General	30	34	64

5. DISCUSSION

A detailed confusion matrix demonstrates the following alarming truth: of 18 authentic articles labeled as 'Information Technology', 17 of them were mistakenly assigned to some other class – mostly, to 'Important Information'. Similarly, nearly half of the authentic 'General' news were automatically assigned to the same problematic 'Important Information' class. Qualitative analysis of such texts confirms that the main reason for this issue is an extraordinary semantic and vocabulary overlap. For example, massive launches of any technological innovations (like a new tax app) are inevitably formulated in extremely urgent, formal and authoritative language which is absolutely identical to important policy statements. Thus, because of this linguistic parallelism, the probabilistic algorithm strongly prefers to use the majority 'Important' label instead of the more specific 'Technology' one.

Full-Stack System Integration and Real-World Deployment

Although the described above limitations of the algorithm on identifying minority classes were revealed, the overwhelmingly outstanding ability of the model to accurately detect important, priority announcements makes its practical application absolutely beneficial. In order to deploy the developed computational model from the experimental Python environment (Jupyter) to the production-grade web environment (ASP.NET Core / C#) of the agency, the whole Keras-trained neural network was serialized into the Open Neural Network Exchange (.onnx) format. Along with the whole model, the crucial auxiliary objects like fitted Tokenizer dictionary and Label Encoder were exported to provide the exact replication of the Python preprocessing math [30].

Table 9. Full-Stack Integration Flow of the AI Recommendation Feature

Workflow Stage	Process Action	System Architectural Description
1. Data Input	News Entry GUI	The Administrator inputs the raw title and body text of the news via the web graphical user interface.
2. Processing	Text Sanitization	The C# backend executes regex cleaning and tokenization, exactly mimicking the Python training configuration.
3. AI Prediction	ONNX Inference	The loaded .onnx model ingests the numerical tensor and computes the Softmax probability distribution.
4. Data Output	Display Recommendation	The ASP.NET dashboard dynamically renders and suggests the highest-probability category on-screen.
5. Human Validation	Administrative Decision	The human Administrator reviews the text, accepting the AI's suggestion or manually overriding the dropdown.
6. Final Storage	Database Commit	The finalized, validated news text and category tag are permanently committed to the

Workflow Stage	Process Action	System Architectural Description
		secure Oracle Database.

The designed solution architecture deliberately uses the 'Human-in-the-Loop' (HITL) approach. The neural network acts as an intelligent advisor never as an autonomous decision-maker. When a new document is created by a PR staff member, the system quickly. Shows the suggestion for the category assignment, on the user interface. However the human administrator can choose to approve the machine suggestion or change it based on his own understanding of the situation. This combined method helps speed up the publication process while making sure that the information stored in the Oracle database is completely accurate [31].

The overall accuracy of 77.78% and weighted F1-score of 0.78 must be interpreted alongside the considerably lower macro-averaged F1-score of 0.4434, which better reflects performance across all three classes equally rather than being dominated by the majority class. This gap is consistent with prior findings that overall accuracy is a misleading metric under class imbalance, and that macro-averaged metrics should be prioritized when minority-class performance carries practical importance [29].

Compared to Transformer-based approaches reported in the Indonesian NLP literature — which have reported considerably higher and more balanced performance on Indonesian news and sentiment classification tasks when trained on larger, more balanced corpora [12] — the present model's weakness on the "Information Technology" class (F1-score 0.069, with 17 of 18 test articles misclassified) is substantially more severe than typical minority-class degradation reported elsewhere. Two dataset-specific factors likely contribute to this. First, the training and validation data were sampled from a merged pool that was artificially and perfectly balanced across categories (3,000 samples each), whereas the authentic test distribution is heavily skewed toward "Important Information" (431 of 513 samples) with very few genuine "Information Technology" examples (only 18). This mismatch between a balanced training distribution and a highly imbalanced real-world distribution likely limits how well the model generalizes to the true rarity and phrasing of minority-class articles. Second, as the confusion matrix in Table 8 shows, IT-related announcements in this institutional context (e.g., a new tax-payment application) are frequently phrased using the same urgent, formal register as high-priority policy announcements, creating genuine semantic overlap that a moderately sized LSTM model, trained on limited authentic examples, struggles to resolve.

These findings carry two practical implications for the agency. First, the current model should not be trusted to independently flag "Information Technology" content, and the human-in-the-loop design described above is not merely a safety margin but an operational necessity given this specific weakness. Second, targeted collection of authentic "Information Technology" articles — rather than further architectural tuning — is likely to yield the largest performance gains, since the core limitation appears to stem from the mismatch between training and real-world class distributions rather than from a fundamental limitation of the LSTM architecture itself.

Future work should prioritize expanding the authentic "Information Technology" training data specifically, so that the training distribution more closely reflects the true, imbalanced distribution encountered in production. Applying class-imbalance-aware training techniques such as oversampling or focal loss, which have shown measurable benefit in comparably imbalanced text classification settings [32], represents a promising direction, as does periodically re-evaluating whether a Transformer-based approach becomes viable once the authentic dataset — particularly for the minority classes — grows substantially larger. Finally, building an automatic retraining loop based on administrator corrections, as originally proposed, remains an important next step for closing the gap between the model's current majority-class strength and its minority-class weaknesses over time.

The full-stack integration of the model into the agency's existing dashboard, described above, demonstrates that a modest, CPU-trainable LSTM model can still be operationally useful despite its minority-class limitations, provided it is deployed within a human-in-the-loop workflow rather than as an autonomous classifier [31].

6. CONCLUSION

This research project set out to build a new system capable of automatically classifying content, using a type of deep learning program known as Long Short-Term Memory. The system was developed for a government agency that handles regional revenue, which had been struggling with a very real problem: every single day, staff had to sort through a large volume of digital documents entirely by hand. To train this system, the research drew on a wide range of information, including the agency's own historical documents, additional material gathered from the internet, and some synthetic data generated with the help of AI. The Long Short-Term Memory model was trained on this combined dataset so that it could learn

to make predictions on its own. Today, the agency's public portal is already using this new system in its day-to-day operations. The rigorous empirical analysis confirmed that the LSTM model performs strongly on the majority "Important Information" class, achieving an overall accuracy of 77.78% and a weighted F1-score of 0.78, with precision rates approaching 90 percent for this class. However, this aggregate figure conceals substantial limitations: the macro-averaged F1-score of only 0.4434, and the near-complete failure to correctly classify "Information Technology" articles (F1-score 0.069, with 17 of 18 test articles misclassified), indicate that the model is not yet reliable across all three target categories. As such, its current production value lies specifically in accelerating and standardizing the classification of high-priority institutional announcements, not in fully automating the classification workflow. Despite these classification limitations, the model's subsequent serialization to the ONNX format and full-stack integration with the agency's existing management dashboard was successful [30], demonstrating that a human-in-the-loop deployment remains a practical path to real-world value even while classification accuracy on minority classes continues to improve. Running continuously in a secure human-in-the-loop system architecture, the implemented machine learning feature currently serves as a smart and rapid recommendation engine. It helps to reduce the cognitive and administrative strain on the employees and greatly standardize complex categorization processes without entirely eliminating human oversight [31]. Future versions of the algorithm and future research regarding this particular system need to concentrate primarily on augmenting the authentic dataset corpus, specifically in regard to underrepresented technological topics. Additionally, it will be vital to develop an automatic continuous re-training loop based on the mistakes made by human administrators while working with the algorithm. In broader terms, the findings of this research suggest that deep learning based classification tools can be meaningfully integrated into the daily operations of regional government institutions, provided that such tools are deployed within a carefully designed human oversight framework rather than as fully autonomous decision-making systems, an approach that may serve as a useful reference for other local government agencies considering similar digital transformation initiatives in their own publication management workflows.

BIBLIOGRAPHY

[1] I. Mutambik *et al.*, "Usability of the G7 Open Government Data Portals and Lessons Learned," *Sustainability*, vol. 13, no. 24, p. 13740, Dec. 2021, doi: 10.3390/su132413740.

[2] S. Sheoran, S. Mohanasundaram, R. Kasilingam, and S. Vij, "Usability and Accessibility of Open Government Data Portals of Countries Worldwide: An Application of TOPSIS and Entropy Weight Method," *International Journal of Electronic Government Research*, vol. 19, no. 1, pp. 1–25, Apr. 2023, doi: 10.4018/IJEGR.322307.

[3] K. Taha, P. D. Yoo, C. Yeun, and A. Taha, "A Comprehensive Survey of Text Classification Techniques and Their Research Applications: Observational and Experimental Insights," *Computer Science Review*, vol. 54, p. 100664, Nov. 2024, doi: 10.1016/j.cosrev.2024.100664.

[4] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning--based Text Classification: A Comprehensive Review," *ACM Comput. Surv.*, vol. 54, no. 3, p. 62:1-62:40, Apr. 2021, doi: 10.1145/3439726.

[5] C. Liu, "Long short-term memory (LSTM)-based news classification model," *PLoS ONE*, vol. 19, no. 5, p. e0301835, May 2024, doi: 10.1371/journal.pone.0301835.

[6] Y. Rafat, P. Narayana, R. M. Mohana, and K. Srilatha, "LSTM-Based News Article Category Classification," *Computer Sciences & Mathematics Forum*, vol. 12, no. 1, p. 8, 2025, doi: 10.3390/cmsf2025012008.

[7] A. Ezen-Can, "A Comparison of LSTM and BERT for Small Corpus," 2020, *arXiv*. doi: 10.48550/ARXIV.2009.05451.

[8] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[9] Z. R. K. Rostam and G. Kertész, "Advances in Pre-trained Language Models for Domain-Specific Text Classification: A Systematic Review," *ACM Trans. Intell. Syst. Technol.*, vol. 16, no. 6, pp. 1–41, Dec. 2025, doi: 10.1145/3763002.

[10] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," *PLoS ONE*, vol. 15, no. 5, p. e0232525, May 2020, doi: 10.1371/journal.pone.0232525.

[11] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? A

comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Information Systems*, vol. 121, p. 102342, Mar. 2024, doi: 10.1016/j.is.2023.102342.

[12] D. Y. Yefferson, V. Lawijaya, and A. S. Girsang, "Hybrid model: IndoBERT and long short-term memory for detecting Indonesian hoax news," *IJ-AI*, vol. 13, no. 2, p. 1913, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1913-1924.

[13] E. Sirait and J. Ismail, "Comparative Evaluation of IndoBERT-Based Architectures for Imbalanced Indonesian News Title Classification," *SinkrOn*, vol. 10, no. 3, pp. 1896–1907, Jul. 2026, doi: 10.33395/sinkron.v10i3.16209.

[14] P. F. Jacobs, G. M. de B. Wenniger, M. Wiering, and L. Schomaker, "Active learning for reducing labeling effort in text classification tasks," Nov. 03, 2021, *arXiv*: arXiv:2109.04847. doi: 10.48550/arXiv.2109.04847.

[15] H. Dai *et al.*, "AugGPT: Leveraging ChatGPT for Text Data Augmentation," 2023.

[16] X.-S. Hong, J.-J. Lee, S.-H. Wu, and M. T.-J. Jiang, "CYUT at the NTCIR-16 FinNum-3 Task: Data Resampling and Data Augmentation by Generation," 2022.

[17] Z. Li, H. Zhu, Z. Lu, and M. Yin, "Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10443–10461. doi: 10.18653/v1/2023.emnlp-main.647.

[18] E. Herrewijnen, D. Nguyen, F. Bex, and K. van Deemter, "Human-annotated rationales and explainable text classification: a survey," *Front. Artif. Intell.*, vol. 7, May 2024, doi: 10.3389/frai.2024.1260952.

[19] S. Choi, J. Sim, and G. Choi, "Synthetic Text as Data: On Usefulness and Limitations," *Applied Sciences*, vol. 15, no. 10, p. 5460, Jan. 2025, doi: 10.3390/app15105460.

[20] M. A. Brown, A. Gruen, G. Maldoff, S. Messing, Z. Sanderson, and M. Zimmer, "Web scraping for research: Legal, ethical, institutional, and scientific considerations," *Big Data & Society*, vol. 12, no. 4, p. 20539517251381686, Dec. 2025, doi: 10.1177/20539517251381686.

[21] A. Luscombe, K. Dick, and K. Walby, "Algorithmic thinking in the public interest: navigating technical, legal, and ethical hurdles to web scraping in the social sciences," *Qual Quant*, vol. 56, no. 3, pp. 1023–1044, Jun. 2022, doi: 10.1007/s11135-021-01164-0.

[22] "Bapenda Jatim." Accessed: May 10, 2026. [Online]. Available: https://bapenda.jatimprov.go.id/?utm_source=chatgpt.com

[23] "Website Resmi Bapenda Madiun." Accessed: May 10, 2026. [Online]. Available: https://bapenda.madiunkota.go.id/

[24] "Teknologi Informasi – Badan Pendapatan Daerah Kota Padang." Accessed: May 10, 2026. [Online]. Available: https://bapenda.padang.go.id/?cat=7

[25] "SIMANTAP - Sistem Informasi Manajemen Pendapatan Daerah Kota Pasuruan." Accessed: May 10, 2026. [Online]. Available: https://bapenda.pasuruankota.go.id/?utm_source=chatgpt.com

[26] "BAPENDA KOTA MALANG." Accessed: May 10, 2026. [Online]. Available: https://bapenda.malangkota.go.id/?utm_source=chatgpt.com

[27] "Bapenda Kabupaten Blitar – Pemerintah Kabupaten Blitar." Accessed: May 10, 2026. [Online]. Available: https://bapenda.blitarkab.go.id/

[28] "Bapenda Kab. Mojokerto Kab Mojokerto." Accessed: May 10, 2026. [Online]. Available: https://bapenda.mojokertokab.go.id/?utm_source=chatgpt.com

[29] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci Rep*, vol. 14, p. 6086, Mar. 2024, doi: 10.1038/s41598-024-56706-x.

[30] H. Heymann, H. Mende, M. Frye, and R. H. Schmitt, "Assessment Framework for Deployability of Machine Learning Models in Production," *Procedia CIRP*, vol. 118, pp. 32–37, Jan. 2023, doi: 10.1016/j.procir.2023.06.007.

[31] S. Amershi *et al.*, "Guidelines for Human-AI Interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, in CHI '19. New York, NY, USA: Association for Computing Machinery, Mei 2019, pp. 1–13. doi: 10.1145/3290605.3300233.

[32] S. F. Taskiran, B. Turkoglu, E. Kaya, and T. Asuroglu, "A comprehensive evaluation of oversampling techniques for enhancing text classification performance," *Sci Rep*, vol. 15, no. 1, p. 21631, Jul. 2025, doi: 10.1038/s41598-025-05791-7.