Adverse Drug Reaction Detection with Drug Prediction Using BERT–LSTM–SVM

Spandana Shivanadhuni & R. Vijaya Prakash

Department of Computer Science and Artificial Intelligence,
SR University, Ananthasagar, Hasnparthy, Warangal, Telangana, 506371, India
*E- mail: research.spandana@gmail.com

Abstract. Social media can be a useful resource for assessing patient-reported drug experiences. Adverse drug reactions (ADRs) are an important public health issue, but finding social media posts in which adverse reactions are described is a difficult task, since not all posts are related to ADR. The proposed work in this research is a hybrid model of BERT-LSTM-SVM for detection of ADRs and drug classification. The input is preprocessed as follows: the text is cleaned, tokenized, normalized, and noise is removed. Bidirectional Encoder Representations from Transformers (BERT) is used to produce contextual representations of the text, and a Long Short-Term Memory network (LSTM) is used to model sequential dependencies to aid binary classification of ADR and non-ADR text. Then, a Support Vector Machine (SVM) classifier is used to classify the associated medicine based on the extracted textual features. In our validation of the method, the accuracy, sensitivity, specificity and precision were 0.99 with a recall. The proposed framework combines contextual language representations, sequential modeling, and supervised classification approaches to facilitate the automatic detection of ADRs and medicine identification.

Keywords: *adverse drug reactions; BERT; long short-term memory; natural language processing; support vector machine; treatment recommendation.*

1 Introduction

Pharmacovigilance is defined as "the science and activities relating to the detection, assessment, understanding, and prevention of adverse effects or any other drug-related problem." This description is based on the World Health Organization's definition of the term. It is made up of postmarketing safety monitoring initiatives that are intended to monitor pharmaceutical benefit–risk ratios and discover new probable adverse drug reaction signals in real-life situations [1]. After a medicine has been approved, ADRs need to be monitored, as they can become a burden on health systems, since they have the potential to result in hospitalization or even death [2]. In addition, alternative sources of pharmacovigilance have also been taken into consideration, such as the use of electronic health information for secondary purposes [3]. After a comprehensive analysis of 25 prospective observational studies, it was determined that 5.3% of patients had been experiencing ADRs [4]. The occurrence of adverse events is

Received on January 22nd, 2026, Revised on May 30th, 2026. Accepted for publication on June 11th, 2026.
Copyright © 2026 Published by IRCS-ITB, ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2026.20.3.7

often the decisive factor in situations when several treatment alternatives are available [5]. Adverse occurrences that are considered to be severe, such as those that result in hospitalization, disability, or death, are not the only significant ones; those that are irritating or interfere with an individual's quality of life are also significant [6]. When it comes to pharmacovigilance data, clinical trials, and other kinds of investigations, it is particularly clear that there is either inadequate reporting or excessive reporting.

Social media platforms are now underutilized while being a significant source of relevant data. Previous research has discovered that social media may be a useful tool for identifying posts that include information about undesirable occurrences [7]. Few studies, on the other hand, have evaluated the value of adverse event reports in social media in comparison to other sources, such as regulatory data, drug information databases (DIDs) that are utilized when prescribing decisions are being made, and other traditional sources [8]. One possible explanation for this is that it is difficult to compare data formats that are different from one another. For instance, research on adalimumab provided a comparison of Twitter data with regulatory data, DIDs, and systematic reviews [9-10].

The main contributions of the present work are:

1. The study proposes a BERT-enhanced LSTM architecture for binary classification, effectively distinguishing between medication-related sentences that represent normal experiences and those that indicate adverse drug reactions.
2. The framework additionally integrates an SVM-based drug prediction module that identifies the most probable medication associated with the detected adverse reaction, enabling end-to-end ADR detection and drug inference from raw text reviews.

The proposed framework can be contrasted with previous transformer-based detection approaches, which mainly only considered the classification of ADR. The proposed framework combines contextual semantic learning, sequential dependency modeling, and medicine prediction in one single architecture. The proposed BERT–LSTM–SVM framework fuses transformer-generated contextual representations, LSTM based temporal feature extraction, and SVM-based medicine classification. The framework is not only capable of detecting ADRs but can also predict drugs based on features derived from user-generated medical reviews. Moreover, the proposed method performs abbreviation expansion, dose normalization, and symptom-aware feature fusion before retrieval, thereby improving its robustness in noisy social media environments. The proposed framework is a combination of these approaches, which sets it apart from the various stand-alone transformer-based ADR detection models.

The remainder of this paper is structured as follows. An overview of existing adverse drug reaction techniques is given in Section 2. The proposed method is presented in Section 3. In Section 4, an extensive experimental findings analysis is provided. Finally, Section 5 concludes the work.

2 Literature Survey

In recent years, the detection of ADRs in social media and biomedical texts has been investigated in the context of pharmacovigilance. Traditional methods typically rely on manually designed linguistic features and statistical learning techniques. However, these methods may fail to detect the semantic relationships and context of patient-generated texts. Thus, deep learning and natural language processing (NLP) have emerged as key methods to detect potential ADRs from unstructured textual information [1-4].

Ying, *et al.* [11] explored how to extract deep linguistic features as well as shallow linguistic features. They identified predicate-ADR relationships by using an ADR lexicon and syntactic dependencies. Other features extracted were then used along with semantic information and part of speech information to build prediction models. Their experiments with Twitter and Daily Strength data confirmed the importance of the integration of various linguistic representations in relation to the detection of ADRs.

Wang, *et al.* [12] proposed a deep neural network method of potential ADR detection. Their approach included the chemical, biological, and biomedical data used with pharmaceutical products. Biomedical literature was represented by word embeddings and relationships between drugs were captured. Another aspect of this approach is the inclusion of a mapping function to deal with medications that are not included in the training set. The results indicated that the integrative use of textual and drug-related data in pharmacovigilance is an approach with potential.

Yousef, *et al.* [13] have proposed a lexicon replacement approach using word embeddings for the extraction of ADRs. They did not change phrases but targeted individual words and matched them with medically oriented word representations. Various classification methods like SVM, Logistic Regression and Naive Bayes were discussed for assessing the extracted features. The importance of proper text representation and text classification techniques in medical text analysis is illustrated in this work.

Zhang and Geng [12] proposed a weakly-supervised CNN/CRNN model for detecting ADEs. They used convolutional neural networks (CNNs) coupled with bidirectional Long Short-Term Memory (Bi-LSTM) networks to learn features at

various levels. The method was designed to lower the noise and the amount of extensive manually labelled training data. These architectures illustrate the potential of deep learning in capturing complementary textual features for identifying ADRs. Zhang et al. [14] presented a multi-hop self-attention mechanism to detect ADRs in social media. The approach they adopted was to extract the contextual information using Bi-LSTM and to focus on different parts of a text using successive attention stages. This approach extracts several lexical characteristics of a sentence, while offering something different from the regular linguistic features.

There have been other studies that explore neural architectures and attention-based approaches for uncovering information related to ADRs from medical reviews and clinical texts [15-18]. Transformer-based methods such as BERT have also been explored to detect adverse drug events and related symptoms [32–35].

The developments shown here indicate increasing attention for contextual language representations in biomedical text processing. In spite of these advances, the detection and characterization of ADRs and identification of the related drugs remain interrelated but separate activities. A system must be able to recognize when a text reports an adverse reaction and which medicine is associated with the adverse reaction. A model suitable for binary classification may not be the right model for detailed analysis of the drug level.

To overcome this issue, this study proposes a hybrid BERT-LSTM-SVM model. BERT produces representations of the context of the text, while LSTM models sequential dependencies. The features generated are then fed to an SVM medicine classification model. A framework that combines contextual representation learning, sequence modelling, and supervised classification is proposed to facilitate the detection of ADR and the identification of the medicine from user-generated texts

3 Proposed Method

The proposed approach provides a complete working framework for detecting ADRs and predicting drugs by using natural language processing (NLP) and machine learning techniques. First, preprocessing is performed on the text, which involves normalization, cleaning, spelling correction, expansion of abbreviations, medical dose normalization and removal of special characters and extra spaces. The cleaned text after preprocessing is tokenized with the BERT tokenizer and fed to BERT to get contextual representations. The BERT–LSTM architecture is able to model the sequential dependency and recognize ADRs from medical reviews provided by users. The framework predicts the drugs for instances

classified as ADR by using an SVM classifier. The classifier is based on [CLS] embeddings obtained from BERT, LSTM features, symptom keywords, and drug-related named entities obtained in the preprocessing stage. The proposed medicine classification workflow is shown in Figure 1. It consists of preprocessing, feature extraction, and classification. The processed text is used to get contextual embeddings, sequential features and domain-specific information. All of these features are provided to the SVM classifier to classify the medicine. The system was evaluated using accuracy, sensitivity, specificity, precision, recall and F1-score. BERT extracts contextual semantic information, LSTM models the sequential dependencies, and SVM makes the classification based on the extracted features. Combining these three dimensions offers a comprehensive approach to detecting and predicting ADRs.

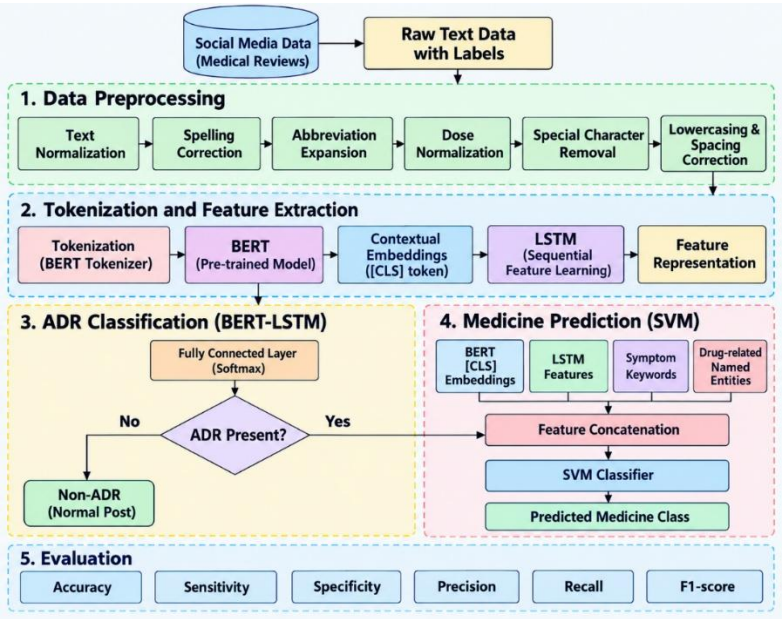


Figure 1 Block diagram of the proposed method.

3.1 Data Preprocessing

The ADR datasets gathered from social media are naturally imbalanced due to the fact that the number of posts that actually contain adverse drug reactions is a small fraction of the size of all drug-related posts. To overcome this, the proposed framework introduces the Synthetic Minority Oversampling Technique (SMOTE) during training to enhance the representation of the minority ADR classes. Moreover, to minimize the bias towards the majority class, class weight

balancing is used for the SVM classifier. These strategies improve the model's performance and the recall performance on minority ADR samples.

The removal of duplicates is a crucial step in preprocessing ADR data to ensure that each adverse event is represented only once in the dataset [16]. First, the criteria for identifying duplicate entries are determined [17-18]. Missing data can affect the analysis and interpretation of the results. In this case, methods such as imputation or deletion are applied [19]. Consistency in data formats, such as date formats or coding schemes, is ensured to facilitate analysis and reduce errors [20]. Tokenization is a fundamental preprocessing step in natural language processing (NLP) that involves breaking down the text into smaller units, called tokens. These tokens can be words, phrases, or other meaningful units of text [21-27]. In word tokenization, the text is split into individual words or tokens based on whitespace or punctuation [28-29]. Common words (stop words) that carry little semantic meaning, such as "the", "and", "of", etc., are removed from the text. Also, rare words with low-frequency counts that are unlikely to contribute to the analysis are eliminated. Spelling errors in textual data are corrected using spell-checking algorithms or dictionaries to reduce noise introduced by typos or misspellings [30].To create high-dimensional word vectors as the model's input, which hold data on the morphology, syntax, and semantics of the words in the word embedding layer, the publicly accessible GloVe toolbox [31] is used to train each word as a k-dimensional word vector, where k denotes the word vector's dimension. After pre-training, the sentence matrix is created by joining the word vectors. Let $W_i \in R^k$ be the *i*-th k-dimensional word vector in a sentence; thus, a drug review with *n*-word vectors is encoded as the sentence matrix $W \in R^{n \times k}$, which is composed of a sequence of word vectors denoted as in Eq. (1)[25]:

$$W = [w1, w2, \ldots.,]T. \quad (1)$$

3.2 Mathematical Model

The mathematical model for ADR detection using a combination of BERT and LSTM consists of two key components: feature extraction (using BERT) and sequence modeling (using LSTM). Let's denote the input text as $X = \{x_1, x_2, \ldots, x_n\}$, [32-38], where x_i represents the *i*-th token in a text comment, and *n* is the number of tokens in the comment. BERT converts the input text into context-aware embeddings through multiple layers of self-attention. Formally, the BERT model processes the input *X* and outputs a sequence of hidden states, as shown in Eq. (2):

$$H_B = \{h_1^B, h_2^B, \ldots\ldots, h_n^B\} \quad (2)$$

where h_i^B is the BERT embedding of the *i*-th token after the final layer of BERT. Each hidden state h_i^B is a high-dimensional vector, typically of size 768 for

BERT-base. Once the contextual embeddings H_B from BERT is obtained, an LSTM is used to model the sequential dependencies between tokens in the text. The LSTM processes the sequence of BERT embeddings and produces a new sequence of hidden states, as shown in Eq. (3):

$$H_L = \{h_1^L, h_2^L, \ldots\ldots, h_n^L\} \tag{3}$$

Figure 2 illustrates the architecture of the BERT-based text classification model. The input tokens, including special markers such as [CLS] and [SEP] [39], are first converted into their corresponding embeddings. These embeddings are then processed through multiple bidirectional transformer layers within BERT, enabling contextual representation learning across the entire sequence. The final contextualized [CLS] token output is passed through a fully connected layer followed by a softmax function to generate the classification result.

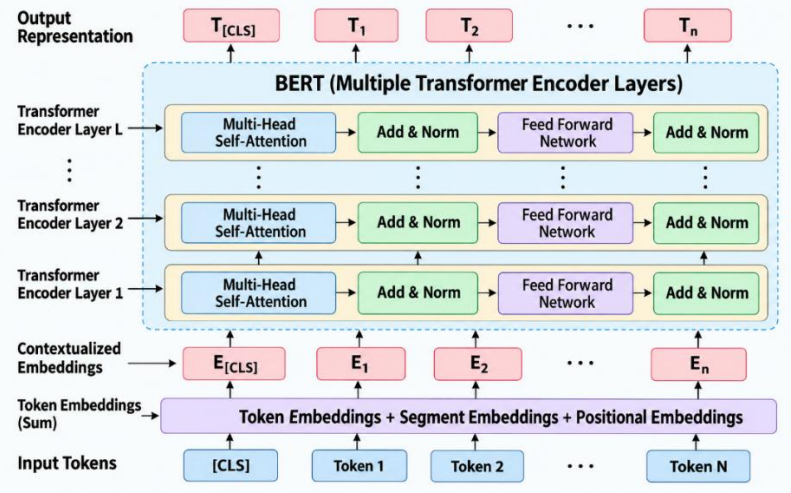


Figure 2 The BERT architecture.

For each time step *t*, LSTM computes the hidden state h_t^L and cell state c_t using the following Eqs. (4) to (9):

$$f_t = \sigma(W_t[h_{t-1}^L, h_t^B] + b_f \tag{4}$$

$$i_t = \sigma(W_i[h_{t-1}^L, h_t^B] + b_i \tag{5}$$

$$o_t = \sigma(W_o[h_{t-1}^L, h_t^B] + b_o \tag{6}$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}^L, h_t^B] + b_c) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{8}$$

$$h_t^L = o_t \odot \tanh(c_t) \tag{9}$$

where f_t, i_t, o_t are the forget, input, and output gates, respectively; W_t, W_i, W_o and b_f, b_i, b_o are the weight matrices and bias vectors learned during training; σ is the sigmoid activation function; and $\odot$ represents element-wise multiplication [40]. The LSTM captures the long-term dependencies between tokens and outputs a new sequence of hidden states H_L, where h_t^L encodes information about the *t*-th token and its relationship to the preceding tokens in the sequence. An optional attention mechanism can be applied on top of the LSTM output H_L to give more importance to the most relevant tokens for ADR detection as in Eq. (10).

$$\alpha_t = \frac{\exp(h_t^L.W_\alpha)}{\sum_{k=1}^{n}\exp(h_k^L.W_\alpha)} \tag{10}$$

where W_α is a learned weight matrix and α_t represents the attention score for the *t*-th token. The final document representation H_{att} is then computed as a weighted sum of the LSTM hidden states, as shown in Eq. (11):

$$H_{att} = \sum_{t=1}^{n} \alpha_t \, h_t^L \tag{11}$$

The final hidden state from the LSTM or the attention-weighted hidden state H_{att} is passed through a fully connected layer for classification. The classification output is computed as shown in Eq. (12):

$$y = \text{softmax}(W_c H\text{att} + bc) \tag{12}$$

where W_c is the weight matrix for the classifier and bc is the bias term. Softmax is used to compute the probability distribution over the output classes. The output *y* is a probability distribution over the classes as presented in Eqs. (13) and (14):

$$y_i = P(\text{Class } i \mid X) \tag{13}$$

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{14}$$

where *C* is the number of classes (two in this case: ADR and non-ADR), y_i is the true label, and $\hat{y}_i$ is the predicted probability for class *i*.

Table 1 presents the parameters that were selected for the proposed BERT model. The vocabulary size is set to 30,522, and the dimensionality of the encoder layers and embeddings is configured to 768. The architecture includes twelve transformer encoder layers and twelve self-attention heads per layer. The model employs the GELU [41] activation function and supports a maximum sequence length of 512 tokens. To enhance generalization, a dropout rate of 0.1 is applied to both hidden layers and attention weights. The model is fine-tuned using a learning rate of 5e-5 and trained for seven epochs.

Table 1 Parameters selected for the proposed BERT model.

Description	Data / Value
Size of the WordPiece vocabulary	**30,522**
Dimensionality of encoder layers and embeddings	**768**
Number of transformer encoder layers	**12**
Number of self-attention heads per layer	**12**
Activation function	**GELU**
Maximum sequence length BERT can handle	**512**
Dropout applied to hidden layers	**0.1**
Dropout applied to attention weights	**0.1**
Fine-tuning learning rate	**5e-5**
Number of training epochs	**7**

Table 2 The hyper-parameters of LSTM.

Description	Data / Value
Number of LSTM units (hidden size)	128 / 256 (commonly used)
Number of LSTM layers	1–2 layers
Activation function for gates	Sigmoid
Activation for candidate cell state	Tanh
Dropout applied to LSTM outputs	0.2 – 0.5
Recurrent dropout applied to recurrent connections	0.2
Optimizer used for training	Adam
Learning rate	1e-3 (typical)
Batch size	32 / 64
Number of training epochs	10–20

In this work, the LSTM configuration was finalized based on such empirical tuning, where commonly used hidden sizes of 128 and 256 units were considered, and the network depth was set to 1–2 LSTM layers, as summarized in Table 2. The LSTM employs a sigmoid activation function for its gating mechanisms and a tanh activation for the candidate cell state. To prevent overfitting, a dropout rate between 0.2 and 0.5 was applied to the LSTM outputs, along with a recurrent dropout of 0.2 for the recurrent connections. The model was trained using the Adam optimizer with a typical learning rate of 1e-3, a batch size of 32 or 64, and 10–20 training epochs.

Figure 3 illustrates the architecture of the proposed LSTM-based classification model. The input sequence, generated from the padded word-level indexing (input size = 55), is processed through 32 LSTM units, each capturing temporal dependencies across the sequence. The hidden representations produced by the LSTM layer are aggregated to form a 256-dimensional hidden layer. These features are then combined through a summation operation and passed to the final output neuron, which yields the predicted class.

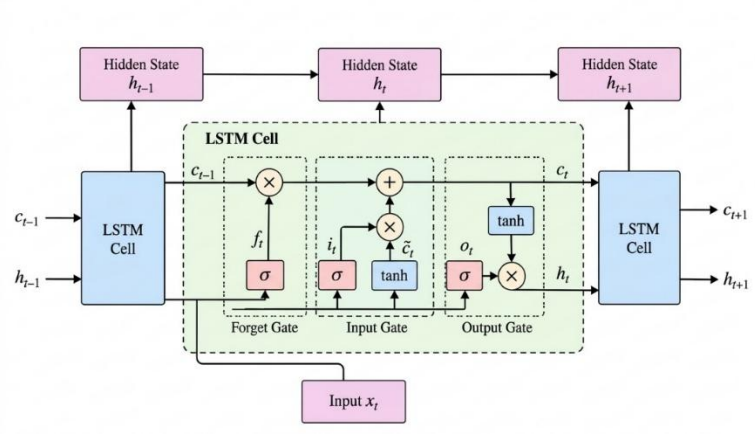


Figure 3 The LSTM architecture.

3.3 Support Vector Machine

SVM is a supervised machine learning algorithm that is widely used for classification and regression. It operates by identifying the optimal hyperplane that maximally separates data samples belonging to different classes within a feature space. The samples positioned closest to this hyperplane are referred to as support vectors and they are essential in determining the orientation and position of the decision boundary.

The decision boundary for binary classification is mathematically represented as in Eq. (15):

$$\mathrm{w} \cdot \mathrm{x} + b = 0 \tag{15}$$

where w denotes the weight vector, b is the bias term, and **x** represents the input feature vector. To achieve an optimal decision boundary, SVM maximizes the margin between the separating hyperplane and the nearest data points. This can be formulated as the following optimization problem in Eqs. (16) and (17):

$$\min_{\mathbf{w},b} \frac{1}{2} \| \mathrm{w} \|^2 \tag{16}$$

subject to the constraint:

$$y_i(\mathrm{w} \cdot \mathrm{x_i} + b) \geq 1, i = 1,2,\ldots,N \tag{17}$$

where $y_i \in \{-1, +1\}$ denotes the class labels and N is the number of training samples.

This formulation ensures maximum margin separation, resulting in robust and accurate classification performance for ADR-related drug prediction tasks. Table 3 presents the detailed numerical configuration of the SVM used for medicine

classification. The parameters include kernel type, regularization strength, gamma value, decision function strategy, and additional optimization settings. These specifications ensure stable convergence and balanced performance across all classes.

Table 3 SVM parameter specifications.

Parameter	Value
Kernel Type	RBF
C (Regularization Parameter)	1.0
Gamma	0.01
Decision Function Shape	One-vs-Rest (OvR)
Kernel Coefficient	Scale
Tolerance	1e-3
Maximum Iterations	10,000
Class Weight	Balanced
Probability Estimates	Enabled
Feature Scaling	Standardization

4 Result and Discussions

The suggested framework was implemented using Windows 10 and an IntelCore i3-8130U CPU running at 2.20 GHz to 2.21 GHz with 8 GB of RAM. The Python programming language was used to implement the framework in Google Colab. The names of 81 medications widely available in the US market were used to capture data for the ADR collection. The tweets included in this collection contained references to various ADRs rather than any particular condition represented by the medications. The data collection comprised training data equaling 75%, and testing data equaling 25%. The implementation details of the proposed framework are briefly summarized here to ensure reproducibility. All experiments were run on Google Colab with Python and TensorFlow and HuggingFace Transformer libraries. A maximum sequence length of 512 tokens was used for the BERT-base uncased tokenizer. The Adam optimizer was used with a learning rate of 5e-5 and a batch size of 32. The hidden layer size of LSTM was set to 128 and the dropout regularization value was set to 0.2. An RBF kernel with balanced class weighting was used by the SVM classifier. Reproducibility of random number generation across experiments was restored by fixing the random seed initialization.

4.1 ADE-Corpus-V2

The ADE-Corpus-V2 Dataset contains information on adverse drug reactions that may be identified. It is a dataset that can be used for classification of whether a statement is associated with an adverse drug event (True) or not (False), as well as for the extraction of the relationship between an adverse drug event and a drug.

The database known as DRUG-AE.rel provides information on the connections that exist between various drugs and the adverse effects they cause. A text file with the name ADE-NEG.txt is used to store all the phrases from the ADE corpus that do not include any adverse effects that are associated with the use of drugs [42-45]. The 13,981 query points that are included in the ADE dataset are separated into two categories: 7000 instances that are not ADE and 6981 instances that are ADE. A Pubmed dataset comprises 32,684 query points, each of which has a reference summary associated with it. Fifty percent of the samples included in this dataset are pertinent to medication-related use case and adverse events. The other samples were disregarded since they are not connected to the problem that we were attempting to resolve.

4.2 Overfitting and Data Leakage Prevention

To avoid overfitting and data leakage, some precautions were taken in the dataset preparation and model training. Before the training, duplicate and close to duplicate records were eliminated. The dataset was divided into training and testing parts in such a way that there was no overlap between the samples before the processing and extraction of features. To maintain class distribution uniformity across the subsets, stratified training–testing splitting was used. In addition, dropout and early stopping were used as regularization techniques for training a better generalization of the model.

4.3 Performance Metrics

The proportion of offensive tweets correctly classified as offensive and non-tweets properly classified as non-offensive (TN) over the entire testing set are examples of accurate measurements, as shown in Eq. (18).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{18}$$

A test's sensitivity describes its capacity to identify people with side effects correctly, as shown in Eq. (19).

$$Sensitivity = \frac{TP}{TP+FN} \tag{19}$$

where TP = true positives, FN = false negatives

Specificity is the capacity of a test to correctly distinguish patients who do not have side effects, as shown in Eq. (20).

$$Specificity = \frac{TN}{TN+FP} \tag{20}$$

The relevance of the findings can be measured using precision, as shown in Eq. (21).

$$Precision = \frac{TP}{TP+FP} \tag{21}$$

Recall assesses the number of actual pertinent results that have been returned, as shown in Eq. (22).

$$Recall = \frac{TP}{TP+FN} \tag{22}$$

There is a reciprocal negative relationship between precision and memory, which can be calculated using the F1 score, which is the harmonic mean of precision and memory,a s shown in Eq. (23):

$$F1\ score = \frac{2*precision*recall}{precision+recall} \tag{23}$$

The performance of the proposed work for second-stage classification showed high accuracy across different stages, as shown in Table 4. In the first stage, ADR/non-ADR classification achieved an accuracy (ACC) of 0.97, sensitivity (SEN) of 0.96, specificity (SPE) of 0.98, precision (PRE) of 0.97, recall of 0.96, and F1 score of 0.98. For the second stage, insulin classification reached 0.98 for accuracy, sensitivity, specificity, and precision, with a recall of 0.97 and F1 score of 0.95.

Table 4 Performance of proposed work for each stage classification.

Performance Metrics		ACC	SEN	SPE	PRE	Recall	F1 score
First stage classification	ADR/Non ADR	0.97	0.96	0.98	0.97	0.96	0.98
Second stage classification	Insulin	0.98	0.98	0.98	0.98	0.97	0.95
	Azithromycin	0.95	0.95	0.94	0.98	0.99	0.98
	Brimonidine	0.92	0.97	0.94	0.99	0.99	0.97
	Amiodarone	0.96	0.98	0.97	0.99	0.98	0.99
	Praziquanttel	0.97	0.98	0.98	0.98	0.99	0.99
	Methimazole	0.98	0.99	0.99	0.99	0.98	0.89
	Pentazocine	0.99	0.98	0.97	0.98	0.95	0.98
	Pentasa	0.98	0.97	0.98	0.98	0.98	0.98
	Flutamide	0.98	0.97	0.98	0.98	0.98	0.99

Azithromycin classification had an accuracy of 0.95, sensitivity of 0.95, specificity of 0.94, precision of 0.98, recall of 0.99, and an F1 score of 0.98.

Brimonidine performed with an accuracy of 0.92, sensitivity of 0.97, specificity of 0.94, precision of 0.99, recall of 0.99, and F1 score of 0.97. Amiodarone achieved an accuracy of 0.96, sensitivity of 0.98, specificity of 0.97, precision of 0.99, recall of 0.98, and F1 score of 0.99. Praziquantel reached an accuracy of 0.97, sensitivity of 0.98, specificity of 0.98, precision of 0.98, recall of 0.99, and F1 score of 0.99. Methimazole classification scored 0.98 for accuracy, 0.99 for both sensitivity and specificity, 0.99 for precision, a recall of 0.98, and an F1 score of 0.89. Pentazocine achieved 0.99 for accuracy, 0.98 for sensitivity, 0.97 for specificity, 0.98 for precision, a recall of 0.95, and an F1 score of 0.98. Pentasa and flutamide both achieved accuracy scores of 0.98, with sensitivity, specificity, and precision also at 0.97-0.98, while recall and F1 scores varied between 0.95 to 0.99.

Table 5 presents the performance of the proposed framework under five different training–testing splits ranging from 70/30 to 90/10. When trained with 70% of the data, the model achieved an accuracy of 0.99, sensitivity of 0.98, specificity of 0.99, precision of 0.99, recall of 0.98, and an F1-score of 0.98. As the training proportion was increased to 75% and 80%, accuracy remained high at 0.985 and 0.982, respectively. Even with reduced training data (85/15 and 90/10 splits), the model maintained strong performance, with accuracy between 0.978 and 0.97, demonstrating excellent generalization capability.

Table 5 Performance evaluation of the model under different training/testing splits.

Train / Test Split	Accuracy	Sensitivity	Specificity	Precision	Recall	F1 Score
70 / 30	0.99	0.98	0.99	0.99	0.98	0.98
75 / 25	0.985	0.975	0.985	0.985	0.975	0.98
80 / 20	0.982	0.97	0.98	0.98	0.97	0.975
85 / 15	0.978	0.965	0.975	0.975	0.965	0.97
90 / 10	0.97	0.96	0.97	0.97	0.96	0.965

Table 6 provides a comparison of the performance metrics between the proposed approach and existing methods for ADR classification. Precision, recall, and F1 scores were evaluated with different methods. The LSTM method referenced from [3] achieved a precision of 0.772, a recall of 0.668, and an F1 score of 0.716. The WSM-CNN-LSTM method from [9] demonstrated improved performance, with a precision of 0.87, a recall of 0.85, and an F1 score of 0.86. The LSTM approach from [10] achieved a precision of 0.82, a recall of 0.86, and an F1 score of 0.84. On the other hand, the CNN method from [11] yielded a precision of 0.56, a recall of 0.84, and an F1 score of 0.85. In comparison, the proposed approach achieved significantly higher performance metrics, with a precision of 0.99, a recall of 0.98, and an F1 score of 0.98.

Table 6 Comparative performance of proposed work.

Methods	ACC	SEN (Recall)	SPE	PRE	F1 score
LSTM [3]	0.73	0.668	0.81	0.77	0.716
WSM-CNN-LSTM [14]	0.86	0.85	0.87	0.87	0.86
LSTM [15]	0.84	0.86	0.82	0.82	0.84
CNN [16]	0.73	0.84	0.59	0.56	0.85
This Work (A1)	0.99	0.98	0.99	0.99	0.98

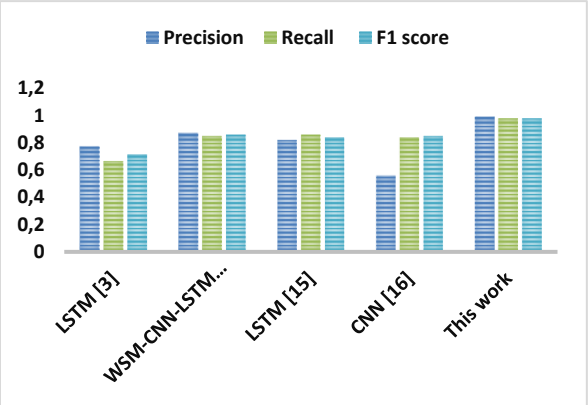


Figure 4 Comparative performance of the proposed method.

A comparative performance analysis of the proposed ADR detection method against existing baseline models in terms of precision, recall, and F1 score is shown in Figure 4. The bar chart demonstrates that the proposed framework consistently outperformed prior approaches across all evaluation metrics. This improvement highlights the model's enhanced capability in accurately identifying ADR-related content. The confusion matrix in Figure 5 illustrates the performance of the proposed ADR detection model across the two classes: ADR and Normal. The matrix shows high true-positive and true-negative counts, indicating strong discriminatory capability.

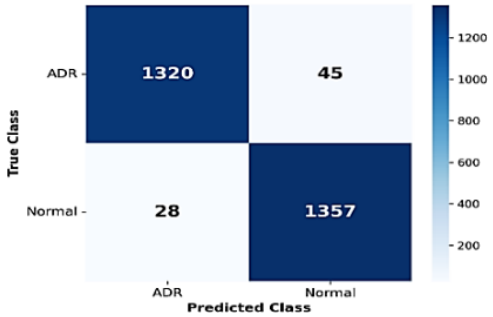


Figure 5 Confusion matrix of ADR detection

The confusion matrix of the proposed model can be seen in Figure 6. The residuals, which are the differences between actual and predicted scores, are mostly centered around zero, suggesting the model is reasonably accurate. However, there is some skewness towards positive residuals, indicating that the model tends to slightly underestimate the severity of some ADRs. The shape of the distribution suggests that while the model performs well for most predictions, there may be certain cases where it fails to capture the full severity of ADRs. Figure 7 shows the training performance of the proposed BERT–LSTM–SVM framework for ADR detection, illustrating the progression of loss, accuracy, and F1 score over 20 epochs. The model shows steadily decreasing training loss alongside improving accuracy and F1 score for both training and validation sets. This trend confirms stable convergence and strong generalization capability.

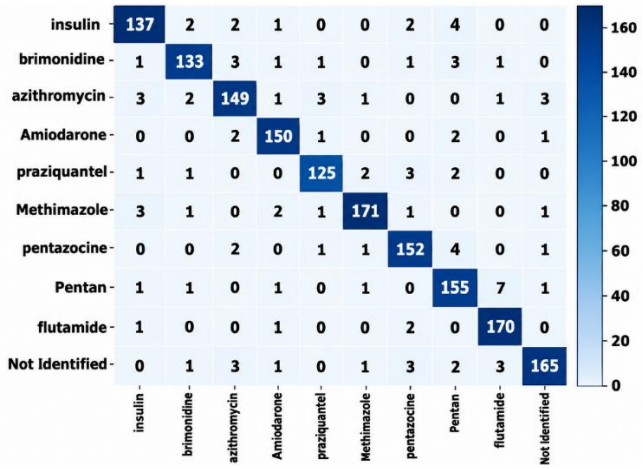


Figure 6 Confusion matrix of ADR medicine classification.

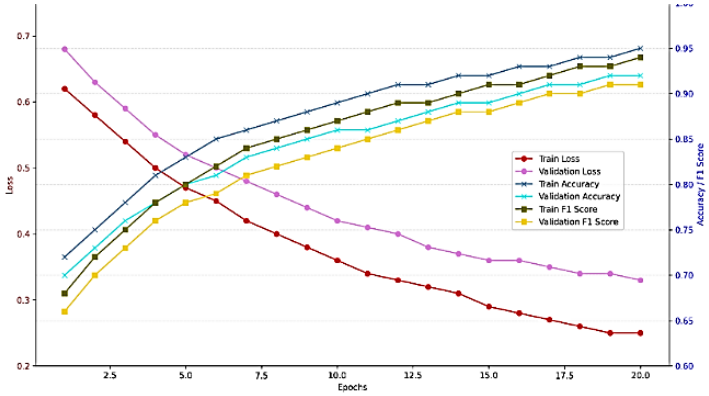


Figure 7 Training performance for ADR detection.

Table 7 presents the 5-fold cross-validation results of the proposed BERT–LSTM–SVM framework for ADR detection. The dataset was divided into five equal folds, where four folds were used for training and one fold was used for testing iteratively. An ablation study was performed to analyze the contribution of each component in the proposed framework. Different model configurations including BERT-only, BERT+LSTM, BERT+SVM, and the complete BERT–LSTM–SVM architecture were evaluated, as shown in Table 8.

Table 7 Five-fold cross-validation results of the proposed method.

Fold	Accuracy	Precision	Recall	F1-Score
Fold 1	0.981	0.983	0.978	0.980
Fold 2	0.984	0.985	0.980	0.982
Fold 3	0.979	0.981	0.976	0.978
Fold 4	0.983	0.984	0.979	0.981
Fold 5	0.980	0.982	0.977	0.979
Average	0.981	0.983	0.978	0.980

Table 8 Ablation study table.

Model	Accuracy	Precision	Recall	F1-Score
BERT	0.94	0.95	0.93	0.94
BERT + LSTM	0.97	0.97	0.96	0.96
BERT + SVM	0.96	0.96	0.95	0.95
BERT + LSTM + SVM	0.99	0.99	0.98	0.98

4.4 Limitations and Future Work

The proposed framework has good performance in detecting ADRs, but there are some limitations. The present study was mainly conducted on English-language data and may not be easily applicable to multilingual social media data. Moreover, the model was tested with only one benchmark dataset and may be difficult to be generalized across platforms. The computational complexity of the hybrid BERT-LSTM-SVM model could also be a problem for deployment in resource constrained settings. Future research will include lightweight transformer architectures, multilingual ADR detection, domain adaptation, and external validation with further pharmacovigilance data.

5 Conclusion

This work presents a comprehensive approach for ADR classification and treatment recommendation using advanced NLP and machine learning techniques. The preprocessing steps, including data cleaning, tokenization, and noise removal, ensure the quality of the input data. The subsequent use of a BERT model for binary classification accurately determines whether a given sentence

expresses a positive or negative reaction towards a medication. Upon identifying positive reactions, an LSTM network predicts the medicine name. The results demonstrate the effectiveness of the proposed approach, with an impressive accuracy of 0.99, sensitivity of 0.99, specificity of 0.99, precision of 0.99, recall of 0.98, and an F1 score of 0.98. This work provides a systematic and efficient approach to ADR classification and treatment recommendation, leveraging state-of-the-art NLP and machine learning techniques. By facilitating informed decision-making in clinical settings, this approach contributes to improving patient care and safety, ultimately addressing one of the most important public health challenges posed by adverse drug reactions.

References

[1] Fan, B., Fan, W., Smith, C. & Garner, H., *Adverse Drug Event Detection and Extraction from Open Data: A Deep Learning Approach*, Information Processing & Management, **57**(1), 102131, 2020.

[2] Rezaei, Z., Ebrahimpour-Komleh, H., Eslami, B., Chavoshinejad, R. & Totonchi, M., *Adverse Drug Reaction Detection in social media by Deep Learning Methods*, Cell Journal (Yakhteh), **22**(3), 319, 2020.

[3] Florez, E., Precioso, F., Pighetti, R. & Riveill, M., *Deep Learning for Identification of Adverse Drug Reaction Relations*, Proceedings of the 2019 International Symposium on Signal Processing Systems, pp. 149-153, 2019.

[4] Alimova, I. & Tutubalina, E., *Detecting Adverse Drug Reactions from Biomedical Texts with Neural Networks*, Proceedings of the 5^{7th} Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pp. 415-421, 2019.

[5] Zhang, T., Lin, H., Xu, B., Ren, Y., Yang, Z., Wang, J. & Duan, X., *Gated Iterative Capsule Network for Adverse Drug Reaction Detection from social media*, 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 387–390, 2020.

[6] McNeil, B.D., MRGPRX2 and Adverse Drug Reactions, Frontiers in Immunology, **12**, 676354, 2021.

[7] Zazzara, M.B., Palmer, K., Vetrano, D.L., Carfì, A. & Graziano, O., *Adverse Drug Reactions in Older Adults: A Narrative Review of the Literature*, European Geriatric Medicine, **12**, pp. 463-473, 2021.

[8] Zucker, I. & Prendergast, B.J., *Sex Differences in Pharmacokinetics Predict Adverse Drug Reactions in Women*, Biology of Sex Differences, **11**, pp. 1-14, 2020.

[9] Magro, L., Arzenton, E., Leone, R., Stano, M.G., Vezzaro, M., Rudolph, A., Castagna, I. & Moretti, U., *Identifying and Characterizing Serious Adverse Drug Reactions Associated with Drug-drug Interactions in a*

Spontaneous Reporting Database, Frontiers in Pharmacology, **11**, 622862, 2021.

[10] Ghasemiyeh, P., Borhani-Haghighi, A., Karimzadeh, I., Mohammadi-Samani, S., Vazin, A., Safari, A. & Qureshi, A.I., *Major Neurologic Adverse Drug Reactions, Potential Drug–drug Interactions and Pharmacokinetic Aspects of Drugs used in COVID-19 Patients with Stroke: A Narrative Review*, Therapeutics and Clinical Risk Management, pp. 595-605, 2021.

[11] Wang, C.-S., Lin, P.-J., Cheng, C.-L., Tai, S.-H., Yang, Y.-H.K. & Jung-Hsien Chiang., *Detecting Potential Adverse Drug Reactions using a Deep Neural Network Model*, Journal of Medical Internet Research, **21**(2), -e11016, 2019.

[12] Zhang, M. & Geng, G., *Adverse Drug Event Detection using a Weakly Supervised Convolutional Neural Network and Recurrent Neural Network Model,* Information, **10**(9), 276, 2019.

[13] Yousef, R.N.M., Tiun, S., Omar, N. & Alshari, E.M., *Lexicon Replacement Method using Word Embedding Technique for Extracting Adverse Drug Reaction*, International Journal of Technology Management and Information System, **2**(1), pp. 113-122, 2019.

[14] Zhang, T., Lin, H., Ren, Y., Yang, L., Xu, B., Yang, Z., Wang, J. & Zhang, Y., *Adverse Drug Reaction Detection via a Multihop Self-attention Mechanism*, BMC Bioinformatics, **20**(1), pp. 1-11, 2019.

[15] Joshi, P., Vedhanayagam, M. & Ramesh, R., *An Ensembled SVM based Approach for Predicting Adverse Drug Reactions*, Current Bioinformatics, **16**(3), pp. 422-432, 2021.

[16] Xiang, W.J., Lu, Z., Qiu, K., Li, P. & Sun, H., *Predicting Drug Risk Level from Adverse Drug Reactions using SMOTE and Machine Learning Approaches*, IEEE Access, **8**, pp. 185761-185775, 2020.

[17] Mantripragada, A.S., Teja, S.P., Katasani, R.R., Joshi, P., V, M. & Ramesh, R., *Prediction of Adverse Drug Reactions using Drug Convolutional Neural Networks*, Journal of Bioinformatics and Computational Biology, **19**(1), 2050046, 2021.

[18] Portelli, B., Passabì, D., Lenzi, E., Serra, G., Santus, E. & Chersoni, E., *Improving Adverse Drug Event Extraction with Spanbert on Different Text Typologies*, International Workshop on Health Intelligence, pp. 87-99, 2021.

[19] Zhang, F., Sun, B., Diao, X., Zhao, W. & Shu, T., *Prediction of Adverse Drug Reactions based on Knowledge Graph Embedding*, BMC Medical Informatics and Decision Making, 21, pp. 1-11, 2021.

[20] Lee, C.Y., & Chen, Y.-P.P., Prediction of Drug *Adverse Events using Deep Learning in Pharmaceutical Discovery*, Briefings in Bioinformatics, **22**(2), pp. 1884-1901, 2021.

[21] Mahendran, Darshini, & Bridget T. McInnes., *Extracting Adverse Drug Events from Clinical Notes*, AMIA Summits on Translational Science Proceedings, 2021, pp. 420, 2021.

[22] Christopoulou, F., Tran, T.T., Sahu, S.K., Miwa, M. & Ananiadou, S., *Adverse Drug Events and Medication Relation Extraction in Electronic Health Records with Ensemble Deep Learning Methods*, Journal of the American Medical Informatics Association, **27**(1), pp. 39-46, 2020.

[23] Magge, A., Tutubalina, E., Miftahutdinov, Z., Alimova, I., Dirkson, A., Verberne, S., Weissenbacher, D. & Gonzalez-Hernandez, G., *Deepademiner: A Deep Learning Pharmacovigilance Pipeline for Extraction and Normalization of Adverse Drug Event Mentions on Twitter*, Journal of the American Medical Informatics Association, **28**(10), pp. 2184-2192, 2021.

[24] Haq, H.U., Kocaman, V. & Talby, D., *Mining Adverse Drug Reactions from Unstructured Mediums at Scale*, In Multimodal AI in Healthcare: A Paradigm Shift in Health Intelligence, pp. 361-375, 2022.

[25] Lee, S., Woo, H., Lee, C.C., Kim, G., Kim, J.-Y. & Lee, S., *Drug_SNSMiner: Standard Pharmacovigilance Pipeline for Detection of Adverse Drug Reaction Using SNS Data*, Scientific Reports, **13**(1), 3779, 2023.

[26] Létinier, L., Jouganous, J., Benkebil, M., Bel-Létoile, A., Goehrs, C., Singier, A., Rouby, F., Lacroix, C., Miremont, G., Micallef, J., Salvo, F. & Pariente, A., *Artificial Intelligence for Unstructured Healthcare Data: Application to Coding of Patient Reporting of Adverse Drug Reactions*, Clinical Pharmacology & Therapeutics, **110**(2), pp. 392-400, 2021.

[27] Nyamabo, A.K., Yu, H. & Shi, J.-Y., *SSI–DDI: Substructure–Substructure Interactions for Drug–Drug Interaction Prediction*, Briefings in Bioinformatics, **22**(6), bbab133, 2021.

[28] Chen, Y., Ma, T., Yang, X., Wang, J., Song, B. & Zeng, X., *MUFFIN: Multi-scale Feature Fusion for Drug–drug Interaction Prediction*, Bioinformatics, **37**(17), pp. 2651–2658, 2021.

[29] Hussain, S., Afzal, H., Saeed, R., Iltaf, N. & Umair, M.Y., *Pharmacovigilance with Transformers: A Framework to Detect Adverse Drug Reactions using BERT Fine-tuned with FARM*, Computational and Mathematical Methods in Medicine, 2021.

[30] Breden, A. & Moore, L., *Detecting Adverse Drug Reactions from Twitter Through Domain-specific Preprocessing and BERT Ensembling*, arXiv Preprint arXiv:2005.06634, 2020.

[31] Nishioka, S., Watanabe, T., Asano, M., Yamamoto, T., Kawakami, K., Yada, S., Aramaki, E., Yajima, H., Kizaki, H. & Hori, S., *Identification of Hand-foot Syndrome from Cancer Patients' Blog Posts: BERT-based Deep-learning Approach to Detect Potential Adverse Drug Reaction Symptoms*, PLOS ONE, **17**(5), pp. e0267901, 2022.

[32] Portelli, B., Lenzi, E., Chersoni, E., Serra, G. & Santus, E., *BERT Prescriptions to Avoid Unwanted Headaches: A Comparison of Transformer Architectures for Adverse Drug Event Detection*, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 1740-1747, 2021.

[33] Heo, E.Y., Jeong, H-j. & Kim, H.H., *Detection of Adverse Drug Reactions using Drug Reviews with BERT+ Algorithm*, KIPS Transactions on Software & Data Engineering, **10**(11), pp. 465-472, 2021.

[34] Qian, J., Qiu, X., Tan, X., Li, Q., Chen, J. & Jiang, X., *An Attentive LSTM based Approach for Adverse Drug Reactions Prediction*, Applied Intelligence, **53**(5), pp. 4875-4889, 2023.

[35] Alshaikhdeeb, A. J., & Yu-N. Cheah., *Utilizing Word Index Approach with LSTM Architecture for Extracting Adverse Drug Reaction from Medical Reviews*, Journal of Advances in Information Technology, **14**(3), 2023.

[36] Wang, X., Wang, X. & Zhang, S., *Adverse Drug Reaction Detection from Social Media Based on Quantum Bi-LSTM with Attention*, IEEE Access, **11**, pp. 16194-16202, 2022.

[37] El-allaly, E.D., Sarrouti, M., En-Nahnahi, N. & El Alaoui, S.O., *A LSTM-based Method with Attention Mechanism for Adverse Drug Reaction Sentences Detection*, In Advanced Intelligent Systems for Sustainable Development (AI2SD'2019), pp. 17-26, 2020.

[38] Grissette, H. & Nfaoui, E.H. *Adversarial LSTM-based Sequence-to-sequence Model for Drug-related Reactions Understanding*, Proceedings of Sixth International Congress on Information & Communication Technology (ICICT 2021), pp. 49-59, 2022.

[39] Wang, Xuqi, Wenzhun Huang, & Shanwen Zhang., *Social Media Adverse Drug Reaction Detection Based on Bi-LSTM with Multi-head Attention Mechanism*, Intelligent Computing Theories & Application: 17th International Conference (ICIC 2021), pp. 57-65, 2021.

[40] Islam, T., Hussain, N., Islam, S. & Chakrabarty, A., *Detecting Adverse Drug Reaction with Data Mining & Predicting its Severity with Machine Learning*, 2018 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pp. 1-5, 2018.

[41] Li, Z., Lin, H. & Zheng, W., *An Effective Emotional Expression & Knowledge-enhanced Method for Detecting Adverse Drug Reactions*. IEEE Access, **8**, pp. 87083-87093, 2020.

[42] Chaichulee, S., Promchai, C., Kaewkomon, T., Kongkamol, C., Ingviya, T. & Sangsupawanich, P., *Multi-label Classification of Symptom Terms from Free-text Bilingual Adverse Drug Reaction Reports using Natural Language Processing*, PLOS ONE, **17**(8), pp. e0270595, 2022.

[43] Sakhovskiy, A., Miftahutdinov, Z. & Tutubalina, E., *KFU NLP Team at SMM4H 2021 Tasks: Cross-lingual and Cross-modal BERT-based Models*

for Adverse Drug Effects, Proceedings of the Sixth Social Media Mining for Health (SMM4H) Workshop & Shared Task, pp. 39-43, 2021.

[44] Tutubalina, E., Alimova, I., Miftahutdinov, Z., Sakhovskiy, A., Malykh, V. & Nikolenko, S., *The Russian Drug Reaction Corpus & Neural Models for Drug Reactions and Effectiveness Detection in user Reviews*, Bioinformatics, **37**(2), pp. 243-249, 2021.