International Journal of Applied Science and Engineering

# Spam classification problems using support vector machine and grid search

**Christine Dewi** [1], **Fransiskus Andika Indriawan** [1], **Henoch Juli Christanto** [2*]

[1] *Department of Information Technology, Satya Wacana Christian University, Salatiga City, 50711, Indonesia*
[2] *Department of Infomation System, Atma Jaya Catholic University of Indonesia, Jakarta 12930, Indonesia*

## ABSTRACT

Spam classification is an important task in identifying unwanted and potentially harmful emails for internet users. The increasing number of internet users highlights the growing importance of handling spam effectively. In this paper, we propose an approach for spam classification using Support Vector Machines (SVM) with grid search hyperparameter optimization. Our research differs from existing studies by specifically focusing on the integration of SVM with grid search to achieve optimal hyperparameter tuning. Additionally, we provide a unique dataset comprising diverse samples of spam emails for evaluation purposes. We also employ pre-processing techniques, including the removal of unnecessary words such as stop words and punctuation marks, as well as word stemming to convert words into their base forms. To optimize the performance of the SVM model, we use Grid Search to determine the optimal values for hyperparameters, including C, gamma, and the kernel. The results of the first experiment using SVM with the first dataset show that grid search yields the optimal parameters {'C': 100, 'gamma': 0.01, 'kernel': 'rbf'}, resulting in an accuracy improvement from 98.02% to 98.47%. In the second experiment using the second dataset, the accuracy obtained is 99.1%, compared to the previous non-optimized parameters which achieved 98.8%. These results indicate a significant improvement in spam classification accuracy. The experimental results demonstrate that our approach outperforms existing methods in terms of accuracy, precision, and recall. The findings of our research have significant implications for improving spam detection systems and enhancing the overall effectiveness of email communication.

***Keywords:*** SVM, Spam classification, Grid search, Machine learning.

**OPEN ACCESS**

**Received:** June 18, 2023
**Revised:** July 11, 2023
**Accepted:** July 21, 2023

**Corresponding Author:**
Henoch Juli Christanto
henoch.christanto@atmajaya.ac.id

**Publisher:**
Chaoyang University of Technology
**ISSN:** 1727-2394 (Print)
**ISSN:** 1727-7841 (Online)

## 1. INTRODUCTION

The number of email users is growing in tandem with the internet's proliferation. Spam, which is caused by unsolicited bulk email messages, is a well-known consequence of email's expanding popularity. As people adapt their daily routines to incorporate the internet, email use is expected to continue increasing. Considered fundamental for communication, email has become the norm. Harmful in nature, spam emails typically contain advertisements. These unwelcome emails are both unopened and unneeded by the recipient. Numerous recipients of email were bombarded by the sender of spam email with an abundance of identical messages. Releasing our email address to deceitful websites or unauthorized parties usually results in the initiation of spam (Sjarif et al., 2019). The adverse impacts of spam are manifold. Among them are slower internet speeds, the loss of significant data, and search engines yielding less accurate results due to the influx of spam content. Spam also leads to unproductive use of valuable time and an overwhelming number of frustrating messages for users. Recognizing spammers and

https://doi.org/10.6703/IJASE.202312_20(4).006

their tactics is pivotal for appropriate countermeasures. Despite extensive research, identifying spam content remains challenging. However, there is still scope for improvement in distinguishing genuine surveys from unsolicited contact attempts (Sultana et al., 2020).

Despite current techniques to identify spam surveys, their efficacy in discerning them is limited. The benefits of removal go unproven, and the varied composition of network elements is unaccounted for. Inefficient communication and high memory consumption impair spam mitigation efforts. Mass email spam and bulk email attacks against people or firms are also common, as are unwanted commercial emails and malicious content-collectively known as spam bot mailing. Such behaviour can seriously harm individuals and groups by gathering personal data, disseminating malware, and influencing public views (Dewi and Chen, 2022).

The main contribution of this research is as follows: (1) To assess the effectiveness of classification algorithms in distinguishing spam emails from legitimate emails, specifically for SVM algorithms coupled with hyperparameter methods using GridSearchCV (Dewi and Chen, 2019). (2) By performing a comparative analysis of these algorithms, this research aims to improve email spam detection and filtering.

The work here is divided into four sections. Our resources and methodology will be detailed in Section 2. Our experimental findings and discussions are presented in Section 3. A summary and suggestions for further research are included in Section 4.

## 2. RELATED WORKS

Spam classification has been extensively studied in the field of machine learning and data mining, with many researchers proposing different methodologies to effectively address the task of accurately identifying and filtering spam emails. In this section, we provide an overview of relevant research that specifically focuses on spam classification using SVM. We emphasize the novelty and significance of our proposed approach, which integrates grid search hyperparameter optimization to enhance the performance of the SVM classifier in spam classification tasks.

SVMs have been widely applied in spam classification due to their ability to handle high-dimensional feature spaces and nonlinear decision constraints. For example, Singh et al. (2018) proposed an SVM-based approach that combines lexical and syntactic features for spam classification. Their research shows the effectiveness of SVM in accurately distinguishing between spam and non-spam emails. Their research also shows that differences in the use of kernels in SVM can also affect accuracy. Although the use of kernels in SVM has an effect, in this study they were able to obtain a high accuracy rate despite using different kernels. However, their work did not explicitly explore hyperparameter optimization.

Furthermore, there have been numerous investigations into the classification of spam utilizing SVMs without explicitly integrating the process of grid search hyperparameter optimization. For instance, a study conducted by Sjarif et al. (2020) explored the implementation of SVM-based approaches that incorporated lexical and syntactic characteristics for the purpose of spam detection. The results of their research proved the efficacy of SVMs in accurately differentiating between spam and non-spam emails. However, their study did not delve into the realm of hyperparameter optimization.

Our research sets itself apart from previous studies by focusing on the originality and importance of spam classification through the utilization of SVM and grid search hyperparameter optimization. By harnessing the capabilities of SVM and meticulously examining the hyperparameter space through grid search, our objective is to discover the most ideal set of hyperparameters that maximizes the effectiveness of classification. This distinctive amalgamation enables the creation of a spam classification model that is both more resilient and precise. Furthermore, our proposed methodology makes a significant contribution to the field by offering a thorough assessment of a vast spam dataset. This dataset consists of a wide range of samples, allowing for a comprehensive evaluation of the effectiveness and applicability of our approach in comparison to existing techniques. By incorporating SVMs with grid search hyperparameter optimization, our objective is to enhance the accuracy, precision, recall, and F1 score of spam classification systems.

## 3. MATERIALS AND METHODS

Fig. 1 describes the workflow diagram in these experiments. The process explains step by step as follows:

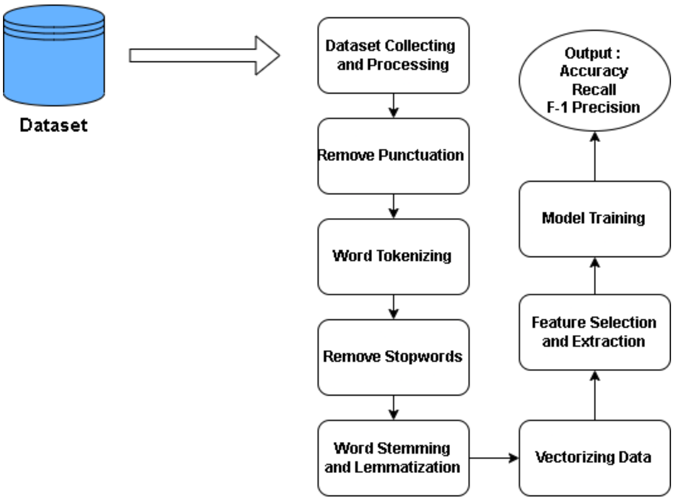
Dataset
Dataset Collecting and Processing
Remove Punctuation
Word Tokenizing
Remove Stopwords
Word Stemming and Lemmatization
Vectorizing Data
Feature Selection and Extraction
Model Training
Output : Accuracy Recall F-1 Precision

**Fig. 1.** Workflow diagram in this experiments

### 3.1 Data Collecting

The first data for this paper was pulled from the UCI machine learning repository. This record contains 5573 pieces of data and 2 types of attributes, where the first attribute contains an indication of spam and the other contains the content of the message. Dataset 1 was taken from the Grumbletext website, manually extracting 425 spam messages sent via SMS. Grumbletext is a forum located in the UK, where mobile phone users openly share their experiences with SMS spam messages (Hidalgo et al., 2006). Most users do not report the spam messages they receive. Identifying the text of spam messages in these claims is a difficult and time-consuming task. It requires careful scanning of hundreds of web pages.

Next, Dataset 1 comes from the NUS SMS Corpus (NSC) dataset consisting of approximately 10,000 valid messages collected for research purposes at the Department of Computer Science at the National University of Singapore. A subset of 3,375 SMS messages were randomly selected from the NSC, mainly consisting of messages sent by Singaporean individuals. In addition, most of these messages were sent by students studying at the university (Cormack et al., 2007). The volunteers who provided these messages were informed that their submissions would be publicly accessible. And 450 ham SMS messages were collected from Caroline Tag's dissertation (Tagg, 2009). For the latter, Dataset 1 was taken from Big's SMS Spam Corpus v.0.1. It has 1,002 SMS ham messages and 322 spam messages (Clarke et al., 2007). The second data used in this article comes from kaggle, which extracts a spam assassin record into a CSV type. The dataset has approximately 30,487 emails which are indicated as spam or ham, but the dataset must be processed first because the dataset is still a txt file (Wander, 2020). The next step is to remove stopwords. In this experiment, NLTK is used to provide an easy way to remove stopwords from the text corpus using the stopwords module. By removing stopwords, we can reduce the noise in the text data and focus on the more important words that convey the meaning of the text (Fayaza and Farhath, 2021).

The reason we use over 30,000 data points is so that the classification algorithm can be trained on a wide variety of texts from the dataset, allowing the model or algorithm to have high accuracy and performance on a wide variety of text samples. Further, Table 1 shows the dataset descriptions in our experiment. Our research employed two datasets namely Dataset 1 and Dataset 2. Besides, the example of Dataset 1 describes in Table 2. Moreover, Dataset 1 consist of 2 features and 5.573 data. Nevertheless, Dataset 2 example shows in Table 3. We classify the data into two labels ham and spam.

**Table 1.** Dataset description

| Dataset | Year | Feature | Amount |
|---|---|---|---|
| Dataset 1 (Tagg, 2009) | 2012-06-22 | 2 | 5.574 |
| Dataset 2 (Wander, 2020) | 2006-06-19 | 2 | 30,487 |

**Table 2.** Dataset 1

| Label | Email |
|---|---|
| Ham | Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat... |
| Ham | Ok lar... Joking wif u oni... |
| Spam | Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's |
| Ham | U dun say so early hor... U c already then say... |

**Table 3.** Dataset 2

| Label | Email |
|---|---|
| Spam | Subject: averatec 3150 h athl 600 + / 256 ram / combo / 12 . 1 " xga / xph @ $<br>910 . 00 only ! ! t o d a y ' s special visit : http : / www . computron - me . com<br>for deals ! 3150 h notebook… |
| Spam | Subject: fw : at our medzplace , you locate top - selling taablets at super decreased<br>prices only… |
| Ham | Subject: thanks for january 1 , 2000<br>i wanted to extend my thanks in advance to all of you who will join me in the<br>office on january 1 , 2000 . i appreciate your willingness to devote part of<br>your holiday weekend to the continuing success story at enron .<br>as a company.. |
| Ham | Subject: re : summer internships at enron<br>vince : thanks . yes it is unfortunate that we were not able to quickly identify who<br>the interested tiger team students were . we will go ahead and process an<br>offer letter for kim and get it to her immediately . also , thanks for agreeing to help out<br>with stanford . hopefully we will get a few good ones ! |

### 3.2 Data Preprocessing

Data preprocessing is essential in this research as it plays a crucial role in the effectiveness and accuracy of the spam classification system. Preprocessing techniques applied to the data, such as tokenization, stemming, and stopword removal, help transform the raw text into a format more suitable for analysis and classification (Chen et al., 2020). In addition, reducing noise and irrelevant information at the preprocessing technique stage can improve the quality of the data and make it easier for classification algorithms to process the data. If data preprocessing is not performed, it will be difficult for the algorithm to process the raw data and the algorithm will produce low accuracy and performance results. This, in turn, can make it difficult for the algorithm to classify the data. In this study, there are several techniques to perform preprocessing in a system (Vijayarani et al., 2016):

- Remove punctuation and stop words from the dataset: stop word elimination is a crucial process in natural language processing that involves removing frequently occurring but insignificant words. Removing stop words is necessary to improve the analysis of a text as they often make the text appear less important and heavier. Eliminating stop words also simplifies the concept space by reducing its dimensionality. Common stop words include articles, prepositions, and pronouns, which have no significant impact on the meaning of a text (Vijayarani et al., 2016).
- Word tokenization: word tokenization is the process of taking a piece of text and breaking it down into individual words, or tokens. In the context of email spam detection, word tokenization can be used to extract individual words from email messages to analyze and identify spam patterns (Shafi et al., 2022). By tokenizing the words in the message, we can analyze the individual words and look for certain patterns, or words, that are commonly associated with spam messages. For example, words like "free," "limited time offer," or "act now" are commonly used in spam messages to encourage recipients to click on a link or provide personal information (Kosasih and Alberto, 2021).
- Word stemming: Stemming refers to the process of generating morphological variations of a root word or a base word. Commonly referred to as stemmers or stemming algorithms, stemming programs are programs that perform stemming. A stemming algorithm will reduce words such as "chocolates," "chocolatey," and "choco" to the root word "chocolate." It will reduce "retrieval," "retrieved," and "retrieves" to the stem "retrieve (Ramasubramanian and Ramya, 2013).
- Word lemmatization: lemmatization is the process by which the different inflections of a word are combined into a single element for analysis. It is like stemming. However, lemmatization provides a context for the words by combining the words that have similar meanings into a single word (Toman et al., 2006). Both stemming and lemmatization are part of the preprocessing of text. However, some people find them confusing and may even use them interchangeably. In fact, lemmatization is often preferred to stemming because it provides a more in-depth morphological analysis of the words that are being processed (Marcińczuk, 2017).

## 3.3 Vectorizing Data

Vectorizing data is the process of converting textual or categorical data into a numerical form that can be easily processed and analyzed by machine learning algorithms. In the context of email spam detection, the vectorization of data involves the conversion of email messages into a set of numerical features that can be used as the input to a machine learning model (Mahajan, 2021).

This technique considers the importance of each word. After the textual data has been vectorized, machine learning algorithms can be used to classify email messages as spam or non-spam based on the numerical features. Logistic regression, naive Bayes, and SVM are some of the most used algorithms for the detection of spam in e-mail messages.

## 3.4 Feature Extraction (Term Frequency- Inverse Document Frequency)

Feature extraction and selection refers to the process of transforming a large, raw data set into a more manageable format by identifying and selecting relevant variables, attributes, or classes. This is a critical step in the training of machine learning models, as it can have a significant impact on the accuracy and reliability of the results (Wan et al., 2019).

The process of feature extraction is the selection of the most important variables which are most representative of the data among the many possible characteristics. This process is called feature selection and is the process of choosing the features to be extracted. Once the relevant features, or variables, have been extracted, they are used to build the model. Proper feature selection is essential for optimal model construction and improved results. In this experiment, we use the Term Frequency-Inverse Document Frequency (TF-IDF) technique to implement the feature extraction technique on the existing dataset (Zareapoor and Seeja, 2015; Cahyani and Patasik, 2021).

TF-IDF is one of the Feature Extraction techniques (). It is a feature extraction technique that determines the importance of a word in a document based on calculations based on numerical statistics. A term's frequency is determined by dividing the number of times a term appears in a document by the total number of words in the text. The formula for calculating Inverse Document Frequency (IDF), which gauges the significance of a term shows in Equation (1) (Imrona et al., 2020):

$$IDF = \log \frac{N}{df_t} \tag{1}$$

TF (t, d) represents the Term Frequency, which measures the frequency of a term (t) in a specific document (d). It can be calculated using different approaches, such as raw term count, logarithmic scaling, or Boolean frequency (indicating the presence or absence of the term). IDF(t) stands for IDF, which quantifies the importance of a term across the entire document collection.

Where:

N is the total number of documents in the collection and DF(t) represents the Document Frequency, which measures the number of documents that contain the term (t). The rationale behind TF-IDF is to give higher weights to terms that are frequent within a particular document but infrequent across the entire collection. This helps to highlight the unique and informative aspects of a document.

By multiplying the TF with the IDF, the TF-IDF score increases for terms that are highly relevant to a specific document while diminishing the importance of terms that are commonly found in many documents.

The TF-IDF values can be used for various purposes, such as text classification, information retrieval, keyword extraction, and document ranking. It helps to identify significant terms within a document collection and allows for effective representation and comparison of documents based on their content.

### 3.5 SVM (Support Vector Machine)

Our experimental approach is based on the SVM, which is a machine learning algorithm that is used to classify and perform linear regression on data. SVM is part of the supervised learning family, this enables it to categorize data into two classes based on observations. With a margin of error, this method generates accurate data maps. SVM is frequently utilized in scientific fields including text categorization, image classification, handwriting recognition, and so on (Ritonga and Purwaningsih, 2018).

The main goals of the SVM algorithm are to categorize any new data that is input. It makes SVM a linear non-binary classification algorithm. SVM algorithm is supposed not only to put an object into a category, but also to provide a safety wide margin between them in the network in the graphics (Assagaf et al.2023). SVM has been implemented in many areas of classification, see the figure, medicine, engineering. SVM has become one of the most popular classification methods because it gives good accuracy in many applied fields. In this study, the author carried out the detection of e-mail spam with the SVM method and observed the performance result (Menaka and Karpagavalli, 2007; Ritonga and Purwaningsih, 2018).

The kernel in SVM is a crucial component of machine learning that employs mathematical functions to convert input data into a feature space with more dimensions. The SVM uses the kernel to classify data efficiently into separate categories by detecting distinctions between them. This process enables accurate classification of nonlinear data (Guenther and Schonlau, 2016). To determine the similarity between data points in the feature space, the kernel function comes to play. In SVM, the radial basis function (RBF) kernel is often utilized to measure similarity based on the distance between two data points. Alternatively, the linear kernel is another frequently used option that determines similarity through evaluating the dot product of two data points (Ibrahim et al., 2021).

SVM facilitates the separation of data into distinct classes by transforming it into a higher-dimensional feature space using a kernel function. The margin between the groups is calculated as the distance between the hyperplane and the nearest data points representing each class. SVM seeks to identify the hyperplane with the most considerable margin as this can effectively accommodate new, unobserved data. Within SVMs there is a kernel trick that offers various facilities, the main reason being the SVM learning process, to determine the support vectors, and only need to understand the purpose of the kernel being used, and there is no need to understand the form of nonlinear equations (Budiman, 2019). Formula Kernel Trick defined in Equation (2).

$$K(x, y) = \Phi(x). \Phi(y) \tag{2}$$

Further, K represents the kernel function, $x$ and $y$ are input data points, and Φ represents the mapping function that transforms the data into a higher-dimensional feature space. By applying the kernel trick, we can directly compute the dot product between the transformed feature vectors Φ(x) and Φ(y) in the higher-dimensional space without explicitly calculating the vectors themselves. This avoids the computational burden of explicitly mapping the data into the higher-dimensional space, which may be computationally expensive or even infeasible for certain types of kernels.

The kernel trick enables SVMs to find nonlinear decision boundaries by using a linear classifier in the higher-dimensional feature space. The SVM optimization problem is expressed in terms of the kernel function, where the decision boundary is defined by a linear combination of kernel evaluations rather than the feature vectors themselves. In essence, the kernel trick allows us to efficiently perform complex calculations in the input space by implicitly operating in a higher-dimensional space defined by the kernel function. This flexibility enables SVMs to handle nonlinear patterns and achieve better classification performance without explicitly transforming the data.

The various varieties of kernel functions are well documented, and a summary of them is listed in Table 4.

**Table 4.** The varieties kernel function

| Type of kernel | Equation | |
|---|---|---|
| Linear | $\mathrm{K}(\vec{\mathrm{x}}_\mathrm{i}, \vec{\mathrm{x}}_\mathrm{j}) = x_i^T x_j$ | (3) |
| Sigmoid | $\mathrm{K}(\vec{\mathrm{x}}_\mathrm{i}, \vec{\mathrm{x}}_\mathrm{j}) = \tanh\left(\alpha\vec{\mathrm{x}}_\mathrm{i}, \vec{\mathrm{x}}_\mathrm{j}\beta\right)$ | (4) |
| RBF | $\exp\left(-\gamma\|x_i - x\|^2\right), \gamma > 0$ | (5) |

#### 3.5.1 Linear Formula SVM

K represents the kernel function. $\vec{\mathbf{x}}_\mathbf{i}$ and $\vec{\mathbf{x}}_\mathbf{j}$ are input feature vectors. $x_i^T$ denotes the transpose of vector $\vec{\mathbf{x}}_\mathbf{i}$. The linear kernel calculates the dot product or inner product between the input feature vectors $\vec{\mathbf{x}}_\mathbf{i}$ and $\vec{\mathbf{x}}_\mathbf{j}$. The dot product operation involves multiplying the corresponding elements of the vectors and summing them. By computing the dot product, the linear kernel measures the similarity or closeness between the feature vectors. If the dot product is high, it indicates that the vectors are more similar or aligned in the feature space. Conversely, a low dot product suggests dissimilarity or misalignment between the vectors.

#### 3.5.2 Sigmoid Formula

K represents the kernel function. $\vec{\mathbf{x}}_\mathbf{i}$ and $\vec{\mathbf{x}}_\mathbf{j}$ are input feature vectors. α and β are hyperparameters that control the scaling and shape of the kernel. The formula computes the dot product between the feature vectors $\vec{\mathbf{x}}_\mathbf{i}$ and $\vec{\mathbf{x}}_\mathbf{j}$ and scales it by α and β. The resulting scaled dot product is then passed through the hyperbolic tangent (tanh) function. The hyperbolic tangent function maps the scaled dot product to a value between -1 and 1, which provides a measure of similarity or nonlinearity between the vectors. The hyperbolic tangent function is commonly used to introduce nonlinearity in the decision boundary of the SVM. The hyperparameters α and β control the scaling and shape of the sigmoid kernel. They determine the steepness and range of the hyperbolic tangent function, affecting the degree of nonlinearity in the decision boundary. Adjusting these hyperparameters allows the kernel to adapt to different data distributions and improve classification performance.

The sigmoid kernel is useful when the data has complex or nonlinear patterns that cannot be effectively separated by a linear decision boundary. By applying the hyperbolic tangent function, the sigmoid kernel allows the SVM to capture more intricate relationships and achieve better classification performance.

### 3.5.3 Rbf Formula

$\vec{\mathbf{x}}_\mathbf{i}$ and $x$ are input feature vectors. γ (gamma) is a hyperparameter that determines the spread or width of the kernel. The formula calculates the squared Euclidean distance between the feature vectors x_i and x, scaled by -γ. The result is exponentiated using the exponential function exp(-z). The RBF kernel measures the similarity or closeness between the feature vectors based on their Euclidean distance. It assigns higher values to vectors that are closer together and decreases the similarity as the distance increases. The scaling factor γ determines the influence of each training sample on the decision boundary, controlling the spread or width of the kernel.

### 3.6 Hyperparameter of SVM

In machine learning, hyperparameters are settings that must be specified before training a model. Hyperparameters are different from model parameters, which are learned during training from the data. One common machine learning algorithm that uses hyperparameters is the SVM (Gul et al., 2021). SVM has several hyperparameters that must be set before training the model. One important hyperparameter is the choice of kernel function, which determines the mapping of the input data into a high-dimensional feature space (Dewi et al., 2021). Other hyperparameters include the regularization parameter (C), which controls the trade-off between minimizing the training error and maximizing the margin, and the kernel coefficient (gamma), which controls the shape of the decision boundary (Wang et al., 2015; Ardhianto et al., 2022).

Grid Search is a technique used for optimizing the hyperparameters of a machine learning algorithm, such as an SVM. This technique involves specifying the range of possible values for each hyperparameter, including a minimum value, maximum value, and number of steps (Syarif et al., 2016). The Grid Search algorithm then creates a grid of all possible hyperparameter combinations and tests each combination using cross-validation to evaluate its performance. By systematically testing all possible combinations, Grid Search aims to identify the optimal set of hyperparameters that will result in the best performance for the model (Hamida et al., 2020; Sulthana et al., 2022). The use of cross-validation helps to prevent overfitting by evaluating the performance of the model on data that was not used for training (Lin et al., 2008; Nugraha and Sasongko, 2022). The goal of Grid Search is to find the best combination of hyperparameters that will allow the model to accurately predict unknown data.

## 4. RESULTS AND DISCUSSION

After Dataset 1 goes through the data preprocessing stage process, the resulting data labeled "spam" means that the message contains or is indicated as spam email, and "ham" means that the message is not indicated as a spam message. This can be seen in Table 5.

**Table 5.** Count the dataset number

| Dataset | Spam | Ham |
|---|---|---|
| Dataset 1 | 747 | 4825 |
| Dataset 2 | 14577 | 15910 |

The model's performance was evaluated by conducting tests on four different models that were created by combining various preprocessing techniques. These techniques included converting words to their base form, removing stop words, eliminating punctuation, and tokenization (Dewi et al., 2022). The following Table and performance chart displays the model that has been tested and produces accuracy, precision, and recall.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{8}$$

According to the Table 6 and Table 7, the algorithm that produces high accuracy is SVM at 98.0% on Dataset 1 and 98.8% on Dataset 2. The second highest algorithm is Logistic Regression which produces 96.2% accuracy on Dataset 1 and 98.4% on Dataset 2. In this case, SVM is proven as one of the algorithms that can classify something and produce quite high accuracy. However, we want to improve the accuracy produced by SVM by using GridSearch on the SVM algorithm.

**Table 6.** Performance model with Dataset 1

| Algorithm | Accuracy (%) | Precision (%) | Recall (%) |
|---|---|---|---|
| SVM | 98.0% | 99% | 86% |
| Logistic regression | 96.2% | 98.2% | 73.5% |
| Random forest | 95.1% | 100% | 64.2% |
| K-nearest neighbors | 92.9% | 100% | 47.6% |

**Table 7.** Performance model with Dataset 2

| Algorithm | Accuracy (%) | Precision (%) | Recall (%) |
|---|---|---|---|
| SVM | 98.8% | 98.1% | 99.4% |
| Logistic regression | 98.4% | 97.3% | 99.4% |
| Random forest | 96.2% | 93.7% | 98.7% |
| K-nearest neighbors | 95.3% | 94.7% | 95.6% |

We use grid search on SVM with parameters 'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['linear', 'rbf', 'sigmoid'] and the amount of training data in the first experiment the training data used was 4,457 and the testing data was 1,115 from Dataset 1. Our second experiment used 26,475 training data and 4,012 testing data on Dataset 2. The results produced after using GridSearch are the right parameters to produce higher accuracy than before using GridSearch (Darmawan and Dianta, 2023). In addition, the reason why the parameters vary in C and Gamma is that when using SVM, C is a parameter that is shared across all SVM kernels, this parameter is very important in terms of balancing the simplicity of the decision surface with the misclassification of the training examples (Chong and Shah, 2022). Lower values of C result in a smoother decision surface, while higher values aim to classify all training examples correctly. Meanwhile, gamma is a parameter that determines the influence a training example has. The larger the gamma value, the closer other examples are to being influenced. Choosing the right values for C and gamma is crucial for the optimal functioning of a SVM (Hussain et al., 2020). To ensure that the selected values are optimal, it is recommended to use GridSearchCV with an exponential distance between potential values for C and gamma (Wainer and Fonseca, 2021). The parameters we get after using GridSearch are {'C': 100, 'gamma': 0.01, 'kernel': 'rbf'}.

After applying the parameters obtained, the accuracy results obtained from the SVM algorithm are 98.47% on dataset 1 and 99.14% on dataset 2. This comparison can also be seen in the following Fig. 2.

Furthermore, our experiment compared the experiment results with research conducted by Poomka et al. (2019). They created a model using deep learning algorithms specifically LSTM and GRU, they also used the same data and compared with the results of research conducted by Almeida et al. (2013). Their experiment compared the results by using the LSTM and GRU models with models using SVM and Naive Bayes algorithms. As a result, the model they created can outperform the model that uses SVM and Naive Bayes. Therefore, we want to improve the accuracy of the SVM algorithm with some data preprocessing techniques on the dataset and hyperparameters on the SVM algorithm (Chen et al., 2020). It can be seen in the following Table 8 that our proposed models and techniques can produce higher accuracy than the LSTM and GRU models, as well as the SVM they compare. Our proposed method the combination of SVM and hyperparameter achieves 98.4% accuracy. This shows that pre-processing and hyperparameter techniques can improve the performance of the model used to classify spam messages in English. Fig. 3 and Fig. 4 is Confusion matrix of SVM with Hyperparameter (GridSearchCV).

**Table 8.** Previous research comparison

| Models | Accuracy (%) |
|---|---|
| SVM (Almeida et al., 2013) | 97.64% |
| NB (Almeida et al., 2013) | 92.05% |
| LSTM (Poomka et al., 2019) | 98.18% |
| GRU (Poomka et al., 2019) | 98.02% |
| SVM + Hyperparameter | 98.47% |

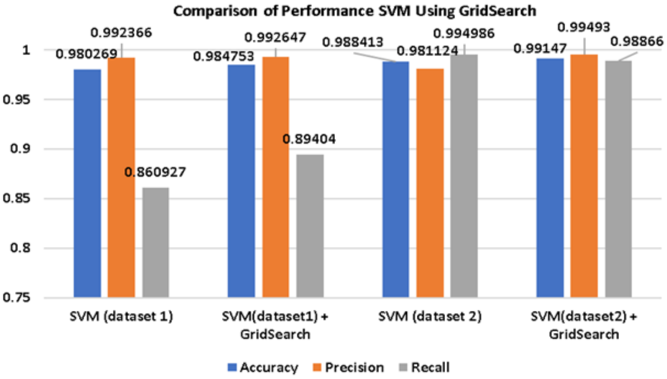

**Fig. 2.** Chart comparison of performance SVM using grid search

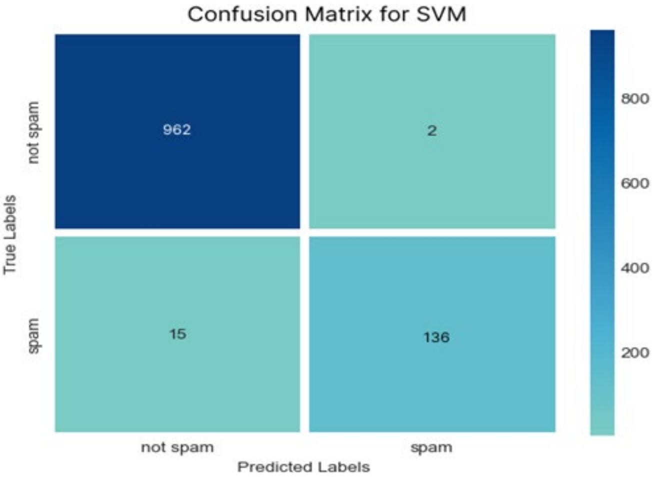

**Fig. 3.** Confusion matrix of SVM with Hyperparameter (GridSearchCV) for Dataset 1

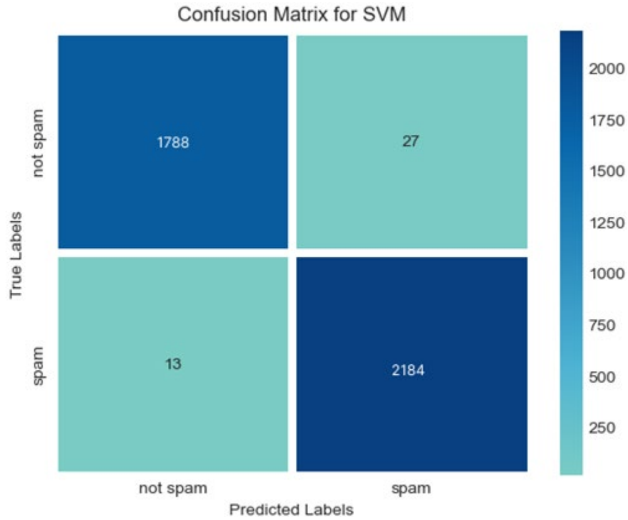

**Fig. 4.** Confusion matrix of SVM with Hyperparameter (GridSearchCV) for Dataset 2

## 5. CONCLUSION

In this research we sort SMS spam with SVM algorithm that uses hyperparameters set by GridSearch, namely, {'C': 100, 'gamma': 0.01, 'kernel': 'rbf'}. To get the model working its best, we use a bunch of techniques for data processing like transformation to base words, ditching the stop words, getting rid of punctuation, and tokenizing. As for testing the model, we used test sets grabbed from not one, but two spam datasets SMS and email. Our results with SVM managed to outperform all other models at 98.47% accuracy on Dataset 1 and 99.14% on the other.

In addition, SVM performance is greatly improved using parameter optimization, although it still works well with default values. The obstacle faced in SVM parameter optimization is that there is no definite range of C and $\gamma$ values. Our belief is that the more diverse the parameters, the better the grid search technique can work in finding the optimal combination of parameters. This makes SVM parameter optimization a very important aspect to ensure maximum success. Using grid search for SVM parameter optimization can significantly improve accuracy, as demonstrated by our experiments, which show that the grid search optimal parameters are generally near the desired range. Nonetheless, grid search has some drawbacks such as being slow and can lead to long execution times.

Therefore, future research based on the proposed approach, there are two main paths to continue this work, improving SVMs and extending hyperparameter optimization to other kernels. Improving SVM: The objective of this study was to provide an uncomplicated version of the learning algorithm for SVM, so as not to disrupt the outcome. There are various methods to advance this research, such as further refining the system to enhance its capabilities. This could be achieved by utilizing other methods to preprocess data, as well as incorporating other techniques that can improve the performance of SVM. Extending hyperparameter optimization to other kernels: SVM employ kernels, such as the Multiquadric Kernel, Polynomial Kernel, and Hyperbolic Tangent to operate on data is our future research.

We used two datasets in this research namely, SMS spam and email spam. Using the datasets we found, we look forward to future research applying SVM combined with gridsearch for real-world applications such as spam filtering across multiple domains. By leveraging SVM with hyperparameter optimization, organizations can develop more accurate and robust spam classifiers. These classifiers have the ability to distinguish between legitimate emails and spam more effectively, thereby reducing the number of false positives and improving the overall efficiency of the spam filtering system. In addition, using SVM with hyperparameter optimization allows for optimal resource utilization on emails. Spam emails consume valuable network bandwidth and computing resources. By effectively filtering spam at an early stage, organizations can optimize resource allocation, leading to reduced storage requirements, improved network performance, and cost savings.

## ACKNOWLEDGMENT

This research would like to thank the support of Satya Wacana Christian University and Atma Jaya Catholic University of Indonesia.

## REFERENCES

Almeida, T., Hidalgo, J.M., Silva, T. 2013. Towards SMS spam filtering: Results under a new dataset. International Journal of Information Security Science, 2, 1–18.

Ardhianto, P., Subiakto, RBR., Lin, C-Y., Jan, Y-K., Liau, B-Y., Tsai, J-Y., Akbari, VBH., Lung, C-W. 2022. A deep learning method for foot progression angle detection in plantar pressure images, Sensors, 22, 2786.

Assagaf, I., Sukandi, A., Abdillah, A.A., Arifin, S., Ga, J.L. 2023. Machine predictive maintenance by using support vector machines. Recent in Engineering Science and Technology, 1, 31–35.

Budiman, E., Lawi, A., Wungo, S.L. 2019. Implementation of SVM kernels for identifying irregularities usage of smart electric voucher. 2019 5th International Conference on Computing Engineering and Design (ICCED), Singapore. 1–5.

Cahyani, D.E., Patasik, I. 2021. Performance comparison of TF-IDF and Word2Vec models for emotion text classification. Bulletin of Electrical Engineering and Informatics, 10, 2780–2788.

Chen, R.C., Dewi, C., Huang, S.W., Caraka, R.E. 2020. Selecting critical features for data classification based on machine learning methods. Journal of Big Data, 7, 52.

Chong, K., Shah, N. 2022. Comparison of naive bayes and

SVM classification in grid-search hyperparameter tuned and non-hyperparameter tuned healthcare stock market sentiment analysis. International Journal of Advanced Computer Science and Applications (IJACSA), 13, 90–94.

Clarke, C.L.A., Fuhr, N., Kando, N., Kraaij, W., De Vries, A.P. 2007. SIGIR 2007. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery, New York, USA.

Cormack, G.V., Gómez Hidalgo, J.M., Sánz, E.P. 2007. Spam filtering for short messages. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 313–320.

Darmawan, Z.M.E., Dianta, A.F. 2023. Implementasi optimasi hyperparameter GridSearchCV pada sistem prediksi serangan jantung menggunakan SVM. Jurnal Ilmiah Sistem Informasi, 13, 8–15.

Dewi, C., and Chen, R.C. 2022. Complement Naive Bayes Classifier for Sentiment Analysis of Internet Movie Database. In Intelligent Information and Database Systems: 14th Asian Conference, Vietnam. 81–93.

Dewi, C., Chen, R.C. 2019. Random forest and support vector machine on features selection for regression analysis. International Journal of Innovative Computing, Information and Control, 15, 2027–2037.

Dewi, C., Chen, R.C., Hendry, Hung, H.T. 2021. Experiment improvement of restricted Boltzmann machine methods for image classification. Vietnam Journal of Computer Science, 8, 417–432.

Dewi, C., Tsai, B.J., Chen, R.C. 2022. Shapley additive explanations for text classification and sentiment analysis of internet movie database. 14th Asian Conference on Intelligent Information and Database Systems, 69–80.

Fayaza, M.S.F., Farhath, F.F. 2021. Towards stop words identification in Tamil text clustering. International Journal of Advanced Computer Science and Applications, 12, 1–6.

Guenther, N., Schonlau, M. 2016. Support vector machines, The Stata Journal, 16, 917–937.

Gul, E., Alpaslan, N., Emiroglu, M.E. 2021. Robust optimization of SVM hyper-parameters for spillway type selection. Ain Shams Engineering Journal, 12, 2413–2423.

Hamida, S., E.L. Gannour, O., Cherradi, B., Ouajji, H., Raihani, A. 2020. Optimization of machine learning algorithms hyper-parameters for improving the prediction of patients infected with COVID-19. 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), 1–6.

Hidalgo, J.M.G., Bringas, G.C., Sánz, E.P., García, F.C. 2006. Content based SMS spam filtering. Proceedings of the 2006 ACM Symposium on Document Engineering, Amsterdam, Netherlands. 107–114.

Hussain, Z.F., Ibraheem, H.R., Alsajri, M., Ali, A.H., Ismail, M.A., Kasim, S., Sutikno, T. 2020. A new model for iris data set classification based on linear support vector machine parameter's optimization. International Journal of Electrical and Computer Engineering, 10, 1079–1084.

Ibrahim, Y., Okafor, E., Yahaya, B. 2020. Optimization of RBF-SVM hyperparameters using genetic algorithm for face recognit. Nigerian Journal of Technology, 39, 1190–1197.

Imrona, M.S., Widyawan, Nugroho, L.E. 2020. Pre-processing task for classifying satire in Indonesian news headline. 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 176–179.

Kosasih, R., Alberto, A. 2021. Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier. ILKOM Jurnal Ilmiah, 13, 101–109.

Lin, S.W., Ying, K.C., Chen, S.C., Lee, Z.J. 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications, 35, 1817–1824.

Mahajan, S.D., Ingle, D.R. 2021. News classification using machine learning. International Journal on Recent and Innovation Trends in Computing and Communication, 9, 873–877.

Marcińczuk, M. 2017. Lemmatization of multi-word common noun phrases and named entities in Polish. Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, 483–491.

Menaka, K., Karpagavalli, S. 2013. Breast cancer classification using support vector machine and genetic programming. International Journal of Innovative Research in Computer and Communication Engineering, 1, 1410–1417.

Poomka, P., Pongsena, W., Kerdprasop, N., Kerdprasop, K. 2019. SMS spam detection based on long short-term memory and gated recurrent unit. International Journal of Future Computer and Communication, 8, 11–15.

Ramasubramanian, C., Ramya, R. 2013. Effective pre-processing activities in text mining using improved porter's stemming algorithm. International Journal of Advanced Research in Computer and Communication Engineering, 2, 4536–4538.

Ritonga, A.S., Purwaningsih, E.S. 2018. Penerapan metode support vector machine (SVM) dalam klasifikasi kualitas pengelasan smaw (shield metal arc welding). Jurnal Ilmiah Edutic: Pendidikan dan Informatika, 5, 17–25.

Shafi, J., Iqbal, H.R., Nawab, R.M.A., Rayson, P. 2022. UNLT: Urdu natural language toolkit. Natural Language Engineering, 1–36.

Singh, M., Pamula, R., Shekhar, S.K. 2018. Email spam classification by support vector machine. In 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 878–882.

Sjarif, N.N.A., Azmi, N.F.M., Chuprat, S., Sarkan, H.M., Yahya, Y., Sam, S.M. 2019. SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. Procedia Computer Science, 161, 509–515.

Sjarif, N.N.A., Yahya, Y., Chuprat, S., Azmi, N.H.F.M. 2020. Support vector machine algorithm for SMS spam classification in the telecommunication industry. International Journal on Advanced Science Engineering Information Technology, 10, 635–639.

Sultana, T., Sapnaz, K.A., Sana, F., Najath, M.J. 2020. Email based Spam Detection. International Journal of Engineering Research & Technology (IJERT), 9, 135–139.

Sulthana, R., Jaithunbi, A.K., Harikrishnan, H., Varadarajan, V. 2022. Sentiment analysis on movie reviews dataset using support vector machines and ensemble learning. International Journal of Information Technology and Web Engineering (IJITWE), 17, 1–23.

Syarif, I., Prugel-Bennett, A., Wills, G. 2016. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. TELKOMNIKA (Telecommunication Computing Electronics and Control), 14, 1502–1509.

Tagg, C. 2009. A corpus linguistics study of sms text messaging. [Doctoral dissertation, University of Birmingham].

Toman, M., Tesar, R., Jezek, K. 2006. Influence of word normalization on text classification. Proceedings of InSciT, 4, 354–358.

Vijayarani, S., Ilamathi, M.J., Nithya, M. 2015. Preprocessing techniques for text mining-An overview. International Journal of Computer Science & Communication Networks, 5, 7–16.

Wahyu Nugraha, A.S. 2022. Hyperparameter tuning pada algoritma klasifikasi dengan grid search. SISTEMASI: Jurnal Sistem Informasi, 11, 391–401.

Wainer, J., Fonseca, P. 2021. How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms. Artificial Intelligence Review, 54, 4771–4797.

Wan, C., Wang, Y., Liu, Y., Ji, J., Feng, G. 2019. Composite feature extraction and selection for text classification. IEEE Access, 7, 35208–35219.

Wander Fernandes. 2020. Enron-Spam dataset, Version 1. Retrieved 2022-12-20 from https://www.kaggle.com/datasets/wanderfj/enron-spam.

Wang, L., Feng, M., Zhou, B., Xiang, B., Mahadevan, S. 2015. Efficient hyper-parameter optimization for NLP applications. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2112–2117.

Zareapoor, M., Seeja, K.R. 2015. Feature extraction or feature selection for text classification: A case study on phishing email detection. International Journal of Information Engineering and Electronic Business, 7, 60–65.