International Journal of Automation and Computing

18(1), February 2021, 55-72
DOI: 10.1007/s11633-020-1247-y

Hybrid Approach to Document Anomaly Detection: An Application to Facilitate RPA in Title Insurance

Abhijit Guha[1,2] Debabrata Samanta[3]

[1] Data Science Department, CHRIST (Deemed to be University), Bangalore 560029, India

[2] First American India Private Ltd., Bangalore 560038, India

[3] Computer Science Department, CHRIST (Deemed to be University), Bangalore 560029, India

Abstract: Anomaly detection (AD) is an important aspect of various domains and title insurance (TI) is no exception. Robotic process automation (RPA) is taking over manual tasks in TI business processes, but it has its limitations without the support of artificial intelligence (AI) and machine learning (ML). With increasing data dimensionality and in composite population scenarios, the complexity of detecting anomalies increases and AD in automated document management systems (ADMS) is the least explored domain. Deep learning, being the fastest maturing technology can be combined along with traditional anomaly detectors to facilitate and improve the RPAs in TI. We present a hybrid model for AD, using autoencoders (AE) and a one-class support vector machine (OSVM). In the present study, OSVM receives input features representing real-time documents from the TI business, orchestrated and with dimensions reduced by AE. The results obtained from multiple experiments are comparable with traditional methods and within a business acceptable range, regarding accuracy and performance.

Keywords: Anomaly detection, title insurance, autoencoder, one-class support vector machine (OSVM), term frequency – inverse document frequency (TF-IDF), robotic process automation, dimensionality reduction.

1 Introduction

The evolution of artificial intelligence (AI) in the past decade has transformed almost all businesses in terms of the way the business processes are handled. A significant paradigm shift has been noticed with operations now being driven by machines in place of human beings, using robotic process automation (RPA). This enables organizations to drive profitability by reducing waste. The complexity of any automation depends on the level of cognitive intelligence required to perform a task. The difficulty increases when the input data takes the form of images, text, document, speech, video, etc.[1–3] which are considered unstructured, in the world of data science. There are growing appeals for automated document management systems (ADMS) that deal with applications such as search, retrieve, profile and classify in the field of healthcare, education, banking, various types of insurances, and other verticals that deal with a voluminous number of transactions. The pictorial representation of a typical ADMS application is shown in Fig. 1.

Title insurance (TI) is a domain of insurance that transacts with documents associated with the property to provide insurance on the title of a subject property to a buyer or lender. Examination of such historical documents that are predominantly legal, needs a very high degree of expertise in the domain. Failing to perform an appropriate examination may lead to flaws in underwriting, resulting in losses in the form of claims to the company. Due to the exhaustive and stringent nature of the examination, it takes somewhere between seven to fifteen days to deliver one title policy. Leading TI companies are on the verge of adopting the AI and machine learning (ML) techniques to automate the examination and underwriting processes to improve the delivery time and quality of an insurance policy by reducing human intervention which in turn has a direct effect on the cost of the policy. To our knowledge, little research to date has been conducted that deals with the ADMS in the domain of TI to improve the RPA capability.

Document anomaly detection (DAD) is one of the important tasks of ADMS. Every business process operates with a set of defined types of documents, which need to be examined by ADMS. However, there is the possibility of receiving an unexpected document in the classification module of ADMS that should be identified and discarded from being sent to the pipeline for any further processing. Various AD techniques, also known as novelty detection methodologies, have evolved in the recent past and been applied to various disciplines such as website management, wireless networks[4], healthcare, network intrusion,

Research Article
Manuscript received April 5, 2020; accepted July 31, 2020; published online October 21, 2020
Recommended by Associate Editor Hui Yu

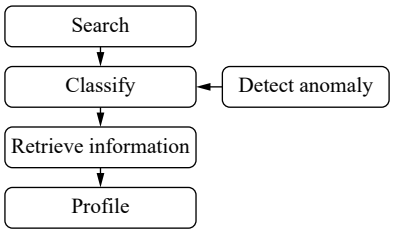

Fig. 1 Typical ADMS flow

web-attack detection, video surveillance, image denoising, fraud detection, etc.[5–9] What makes the task of DAD more difficult in the TI domain is that firstly, the documents have very similar content and secondly the training time needs to be as optimized as possible because there is a need for frequent and constant improvement of the DAD model in ADMS by frequent retraining. For example, it is easier to model the distribution of a positive class of political news against the sports news corpus as a negative class because the two corpora have very distinct words and contexts. But, the task of detecting anomalies by modeling articles on cricket news where the expected negative class articles are football news is difficult as both the articles belong to the sports category and share similar words and context. Other than these complexities which are specific to this domain, there are other generic complexities involved with AD modeling in comparison to traditional binary or multiclass classification modeling[10].

The exclusivity of AD lies in the fact that only normal data (positive class) are considered for modeling. In real-time scenarios, it is difficult or costly to gather the non-positive class samples[5]. For example, a fraudulent banking transaction is a very rare event compared to non-fraudulent transactions. Due to a high imbalance and cost, there arises the need for a single class data modeling which is supposed to identify the abnormality of any anomalous behavior having known the normal pattern. Unlike any binary or multiclass classification modeling which is discriminatory in nature[10, 11], AD modeling is more complex and the complexity grows by manifold when data is multivariate, high-dimensional and sparse, specifically for texts which follow the power-law distribution[12]. The presence of features that are irrelevant can camouflage the anomalies[13–17]. This is known as the curse of dimensionality. A two-stage approach is necessary for a system where in the first stage, the features are engineered[18] to discard the unwanted ones and represent the data in a much more concise and lower-dimensional space with prominent attributes.

Classical approaches of AD do not proclaim consistency with the increased dimensionality[12] of the data[16]. There is a need for machine learning based technology support in the domain of AD to cope with the curse of dimensionality[19]. Although the growing popularity and effectiveness of the deep learning based approaches have been noticed in recent years[19] to solve various problems for image data including AD, little attention has been paid to the AD for text data[12].

In the present study, a hybrid approach for anomaly detection, using unstructured, image-based text documents concomitant to TI has been introduced and compared with the traditional one-class support vector machine (OSVM)[20,21] approach to achieve a model with high performance and accuracy acceptable in a real-time business process framework. A reconstruction-based approach using autoencoders (AE) has been adopted for feature extraction and dimensionality reduction followed by AD using OSVM. Documents are represented in a vector space using TF-IDF that has been considered the input for AE[22]. Error distribution of the reconstruction loss has been scrutinized before applying the statistical measures for measuring the distance of new data points. Experiments have been carried out using multiple combinations of positive classes and compared with results obtained using only OSVM without dimensionality reduction[23]. The hybrid strategy achieved considerably higher accuracy with enhanced performance.

The remainder of the paper is organized into five sections. Section 2 commences with a concise review of the existing literature on AD, different techniques, and its application. The methodology of the research is explained in step by step approach in Section 3. Section 3 also affirms the techniques of data preparation and feature mining adopted for the experiments and presents the result of the exploratory study of the data. All the decisions taken in the methodology are backed by reasoning and those are presented in this section. AE and the architecture adopted for the experiment are explained in Section 4 following which the OSVM based multivariate AD is discussed in the same section. Finally, the results along with the constraints and future scope of the experiments are discussed, analyzed in Section 5 followed by conclusive remarks in Section 6.

2 Background and related study

AD has been established as a technique useful for multiple disciplines over the past decade. Various methods of AD evolved simultaneously with the progress of technologies[6, 24]. A thorough background study has been conducted to explore different domains and techniques of AD algorithms since 2009[6, 25].

2.1 Anomaly detection in various domains

Mahadevan et al.[26] introduced the early incorporation of AD in computer vision on video sequence data representing crowded scenes[27]. This work was continued by them where they proposed localization and spatial, temporal anomaly detection techniques. This research was conducted to facilitate the video surveillance monitoring task. Sabokrou et al.[28] proposed a fully connected

CNN architecture for anomaly detection in crowded scenes to be incorporated in video surveillance.

A study on the intrusion detection system (IDS) was presented by Kim et al.[29] using a hybrid model of two methods of misuse detection and anomaly detection[30]. The objective of the research was to be able to detect attacks in a network system. The misuse detection component was designed to detect the known attacks whereas anomaly detection was used for identifying unknown attacks within the network. A self-supervised approach for network intrusion detection based on a restricted boltzmann machine is proposed by Fiore et al.[31] The authors here brought attention to a crucial point of the ever-changing and evolving nature of network anomalies and the requirement of a self-supervised machine learning-based approach to deal with the dynamic nature of intrusion.

AD is implemented in the application of remote sensing[32]. By studying the hyperspectral imagery object detection, distinguishing is possible based on the spectral signature of the objects. It is a well-known task for a remote sensing community to be able to discriminate uncommon objects from known and common objects. Web search engine accuracies depend highly on the offline component of the engine which consists of a crawler, web graph, and indexing. Anomalies in the web graph component compromise highly on the search indexing and accuracy. Papadimitriou et al.[33] studied five different similarity schemes to identify anomalies arising due to problems during preparation of the offline web graph component. Ten et al.[34] proposed an anomaly detection tool to detect cyber-attacks to a power grid. An exponential increase in the availability of streaming and time-series data has been noticed with the rise of internet of things (IoT) with real-time data sources[35]. The authors proposed a generic anomaly detection mechanism to be used for streaming data across domains.

Schlegl et al.[36] proposed AnoGAN (anomaly detection generative adversarial network) in anomaly detection in imaging data produced in medical imaging for disease diagnosis and treatment responses. System logs are important records of computer system behaviors to deal with critical states as well as system failures. Mining the log information helps in root cause failure analysis by finding anomaly in the log sources. A sequence-based deep learning model is proposed by Du et al.[37] Lu et al.[38] proposed a reinforcement learning-based anomaly detector to be used in unmanned aerial vehicles that are used in farming, weather observation, infrastructure inspection, etc. Malicious user behavior in social networks is one of the major problems in today's world. Identifying unusual behaviors helps to stop unwanted activities. Study and modeling of normal user behavior and identifying anomalies has been studied[39].

Yan and Yu[40] showed a unique way to find anomalies in the field of production and health monitoring (PHM). The authors claimed to have explored a deep learning-based technique in a dire need to improve the system.

2.2 Deep learning techniques in anomaly detection

The above literature shows certain domains where anomaly detection is prevalently used. Network intrusion detection is one among them. Video surveillance, hyperspectral imaging, medical imaging, search engine optimization are other areas where many pieces of research have been noticed in the past decade[41]. There has been significant research and improved application noticed in the area of deep learning[42–45] with the advent of computing power and ever-increasing data size. Naturally, many researchers have put in their efforts to study the applicability of deep learning in the area of anomaly detection.

A hybrid approach of anomaly detection using deep belief networks followed by one-class SVM like the present study has been proposed[46]. A thorough comparative study has been conducted by the researchers to find the best hyperparameters for the proposed hybrid algorithm. The experiments were conducted with high dimensional data gathered from different IoT devices.

An and Cho[47] experimented with the anomaly detection using MNIST and KDD Cup 1999 network intrusion dataset and proved reconstruction probability-based anomaly detection using variational autoencoder that performs better than the reconstruction error-based anomaly detection using the autoencoder. They also considered the principal component-based approach for the comparison and variational autoencoder (VAE) was established to have outperformed all other methods.

Another study proposed by Sakurada and Yairi[23] experimented with an autoencoder for non-linear dimensionality reduction, compared and proved to be performing better than the principle component analysis and kernel principle component analysis approaches of linear reduction of dimensionality. The unique aspect of the study is that the researchers used artificially simulated time-series satellite data as well as real data from spacecraft telemetry.

A unique study showed the applicability of a long-short-term-memory (LSTM) based neural network approach to detect anomalies in the system log[48]. It was experimented on along with traditional approaches of data mining techniques and found to be performing better than those approaches[37].

A contemporary study was conducted in the field of production and health monitoring (PHM) system where deep learning was used to extract the features automatically from multiple sensors of gas turbines to monitor the health of combustors[40]. The authors claimed this study to be the first in the field of PHM. They established the argument concerning the traditional way of handcrafted

feature generation comparing to the deep way of auto-generation. The extreme learning machine (ELM) was used for the anomaly detection task for both the hand-crafted features and features generated using deep mechanism and the result showed a significant improvement in accuracy with the features generated through the deep learning mechanism.

Use of deep learning techniques replacing the traditional models has been noticed in orthodox fields such as hyperspectral imagery[48], anomalous event detection in a video[49], and network intrusion detection[50]. There is a clear dearth of study and utilization of deep learning techniques discerned in the field of large texts.

2.3 Autoencoders in anomaly detection

The AD technique is not an inherent capability of any state-of-the-art deep learning algorithms[51] but for any anomaly detector to perform, the supplied input features are required to be engineered well so that the important features can be identified and retained and at the same time unimportant features can be expelled to deal with the problem of dimensionality. Traditional approaches of dimensionality reduction and feature extraction such as PCA has its limitations, being linear in nature[22]. Various application domains today deal with data that have non-linearly related features with a very high dimension. Autoencoders are one such deep mechanism that apply reconstruction-based mechanisms for nonlinear pattern identification and are applied in diverse data types in recent years for many applications including anomaly detection[22]. With the advent of variations in autoencoders such as variational autoencoders, denoising autoencoders, a stacked autoencoder, we have noticed a growing popularity of reconstruction error and reconstruction probability based anomaly detection in various fields[52].

AD in the field of medical imaging was researched using context-encoding variational autoencoders[53] and was proved to be better performing than the state-of-the-art reconstruction error based anomaly detection. The study was able to identify an abnormal region from a medical image with much higher precision than the state-of-the-art techniques.

Jeragh and AlSulaimi[54] incorporated autoencoder based hybrid techniques of credit card fraud detection. The hybrid mechanism consisted of a layer of autoencoder followed by a standard OSVM. The model showed results comparable with traditional OSVM.

Chong and Tay[55] used an autoencoder based anomaly detector that used spatiotemporal data in the form of video streaming and the accuracy was comparable with state-of-the-art techniques.

2.4 One-class support vector machine for anomaly detection

One-class SVM has been used more than any other algorithms for solving real-time issues pertaining to anomaly detection[56]. It is noticed that the researchers have adopted various techniques to improve the accuracy of OSVM by either plugging the algorithm with another state-of-the-art feature extractor or by improving the algorithm itself[57–59]. In the study, Amer and Goldstein[56] improved the OSVM by suggesting two approaches. They made the outliers impact less on the decision boundary of OSVM and tested the modification using the UCI machine learning data set and one of the approaches showed a promising result.

A hybrid approach to combine a misuse detection and an anomaly detection model is used[29]. At the first step, the misuse detection model was trained using a decision tree and then the multiple anomaly detection model was trained using OSVM to train subsets of known attack information. The proposed model showed better performance than conventional approaches.

Hejazi and Singh[60] compared the performance of a two-class SVM and one-class SVM with credit card fraud data. The study was conducted with different kernels with multiple settings of other parameters using both balanced and unbalanced data sets. The result confirms the superiority of the OSVM over the binary classifier.

Detecting system behavior anomalies at the host level is a challenging task[61]. A high rate of false alarms has been noticed using traditional techniques. The study showed a novel approach of feature extraction and combined that with an OSVM model[62]. The result showed significant improvement in reducing false-positive predictions.

2.5 Other methods to anomaly detection

Chandola et al.[5] adopted to solve anomaly detection problems across various application domains into six groups: classification, clustering[63], nearest neighbor, statistical, information theory and spectral theory based.

Classification based anomaly detectors are further subdivided into four further subcategories and those are:

1) Neural network based techniques: This category consists of multi-layered perceptron, neural trees, auto-associative neural networks, adaptive resonance theory based, radial basis function based, hopfield networks and oscillatory network.

2) Bayesian network based.

3) Support vector machine based.

4) Rule based.

Nearest neighbor based anomaly detection was further categorized into:

1) K-nearest neighbor based approach.

2) Density based approach.

Clustering based anomaly detection were identified to be containing the four below techniques:

1) K-means clustering.

2) Expectation maximization (EM).
3) Self-organization map (SOM).
4) Clustering based local outlier factor (CBLOF).

Traditional statistical AD in the below techniques:
1) Regression model based.
2) Parametric test based.
3) Autoregressive integrated moving average (ARIMA) and autoregressive moving average (ARMA).
4) Kernel function.
5) Non-parametric test based.

3 Methodology

The experiments were conducted with real-time document samples collected from the production data store of two document management systems (DMSs) of a reputed TI company, partially automated using typical RPA. In one business process, the expected document types are purchase and sales agreement (PSA), medium property details report (MPDR) and TAXES and in the other business process, the expected incoming documents to the ADMS are DEED, title commitment (TC) and voluntary lien report (VLR). Document types from the first process were considered as Group 1 and the same from the latter process were considered as Group 2.

The unique need of the ADMS is that, within the workflow of the ADMS, some of the classifiers expect only one document type and some expect a combination of multiple documents. Due to this, it is not enough for the AD to work on a single type of document but also a combination of multiple document types. In traditional AD, it is noticed that the known data points belong to a single population but in the present study, it is also taken into consideration that the known data points or positive class data points may come from multiple populations. Being discriminatory in nature, a regular classifier has limitations to identify an unknown class. It always classifies an unknown observation as one of the classes it was trained on.

These experiments aimed to study the possibility of an AD, associated with an AE component which reduces the dimensionality of the document features without compromising the accuracy of it and improves the overall performance and manageability of the system. Two simultaneous experiments were designed which were carried out in the below stages as shown in Fig. 2. In one experiment, the traditional technique of OSVM was considered, and in the other, one OSVM was accompanied by AE for feature extraction and dimensionality reduction.

Data Collection: Random sampling was done from each class of documents keeping the number proportional to the propensity of the documents in the process which makes the class distribution imbalanced and adds another degree of complexity. Fig. 3 below shows the count distribution of the selected samples of six different document types. The most important component in ADMS is the classification module. This module identifies an incoming document as one of the types the classifier is trained on and passes on to the next module that is the information extractor, etc. A limitation of a classifier is that due to the discriminatory nature[10] of it, it always classifies the incoming document as one of the trained types despite the possibility that the incoming document does not belong to any of the trained types.

Pre-processing: Sample documents were of two formats, image-based portable ducument format and tagged image file format. Text data were extracted from the images using the optical character recognition (OCR) engine. Standard text pre-processing techniques were adopted in the study. Operations such as lowercasing, stopword removal, punctuation removal, numeric values removal, non-ASCII character removal were performed chronologically on the corpus. Document term matrix was prepared. After removing all those tokens that appeared in less than ten documents. It ensured the removal of most of the non-dictionary and gibberish tokens generated due to limitation of OCR for poor quality images.

Exploration: A thorough exploratory analysis was performed to understand the data adequately in the first

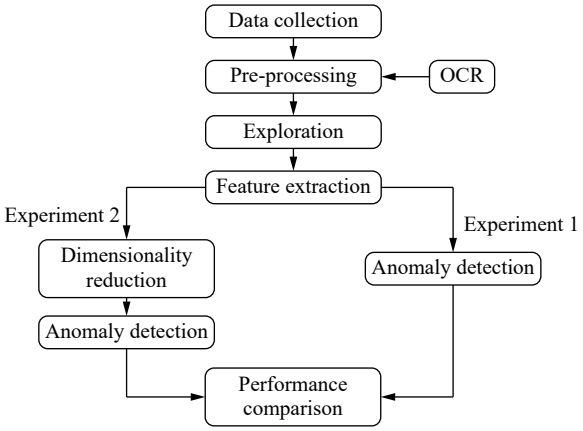

Fig. 2 Stages of experiment

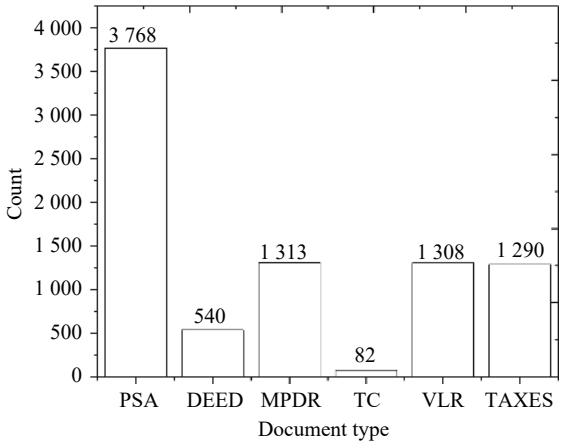

Fig. 3 Count distribution of document types

stage of the experiment. As the present study is more focused on understanding the distribution of the data and concept learning, exploring the data through techniques used in unstructured text data mining was very important. The size and distribution of the tokens associated with each type of documents were explored before representing the documents into an n-dimensional feature space.

Feature extraction: After extracting the text data from the images and performing standard pre-processing tasks, documents were represented in feature space using the two most popular embedding techniques[64]. TF-IDF vectorization and Doc2Vec embedding (derived from Word2Vec)[65] were evaluated for the study. Both the techniques represented the documents in a very high dimensional space which was impossible to visualize in its original form. Projecting the high dimensional space into two-dimensions or three-dimensions was necessary for the visualization of the documents in space.

Eighty documents were sampled randomly from each category and represented in a 2D space using the t-distributed stochastic neighboring embedding (t-SNE)[66] algorithm to visualize the distribution of the documents in a 2D space. The visualization was created for both TF-IDF and Doc2Vec features[66]. Stochastic neighboring embedding converts the Euclidian distance of two points a_i, b_j on a high dimension space into the conditional probability $p_{j|i}$ that represents the similarity. $p_{j|i}$ represents the probability of selecting a_j as neighbor from a Gaussian distribution with mean at a_i. The value of $p_{j|i}$ is relatively low for data points that are separated wider than those points which are closer to each other. Mathematically, $p_{j|i}$ is represented as

$$p_{j|i} = \frac{\exp\left(-\|a_i - a_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|a_i - a_k\|^2/2\sigma^2\right)} \tag{1}$$

where σ_i is the variance of the Gaussian distribution for which the mean is a_i. Considering the value of $p_{i|i}= 0$, $q_{j|i}$ is the conditional probability of the similarity of the same two data points in the lower dimension, $q_{j|i}$ is represented as

$$q_{j|i} = \frac{\exp\left(-\|l_i - l_j\|^2\right)}{\sum_{k\neq i}\exp\left(-\|l_i - l_k\|^2\right)} \tag{2}$$

where variance in the lower dimension is $\frac{1}{\sqrt{2}}$. If the mapping of high dimensional points is correctly done in lower dimension, $p_{j|i}$ and $q_{j|i}$ will be the same. From this observation, the objective of t-SNE is to minimize the difference between $p_{j|i}$ and $q_{j|i}$ using the Kullback-Leibler divergence using gradient descent. The cost function is represented by

$$C = \sum_i KL\left(P_i \| Q_j\right) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}. \tag{3}$$

Figs. 3 and 4 show the 2D representation of the documents and the document classes have a well-separated boundary in 2D space. TF-IDF representation of the features showed a clearer separation boundary than that of Doc2Vec. t-SNE representation for Doc2Vec had an overlapping region of DEED and PSA documents. Learning the distribution of PSA and DEED would be difficult using Doc2Vec because of the overlapping region of both the document types. There were two advantages of using Doc2Vec over TF-IDF. Dimensionality and sparsity could not be controlled in TF-IDF measure. However, in Doc2Vec, the dimensionality could be controlled, and sparsity is nil as mentioned in Table 1.

Based on the visual exploration of the data points, the TF-IDF representation of the documents was chosen over the Doc2Vec for the present study due to a better separation of PSA and DEED classes. In a process where PSA is the only expected positive class, if an incoming DEED document is observed by the AD module, it would be a very difficult job for the AD to be able to detect the incoming document as an anomaly if the documents were represented in the feature space using Doc2Vec.

Architecture 1: Documents were represented in the TF-IDF or Doc2Vec feature space and sent to OSVM for anomaly detection.

Architecture 2: Documents were represented in TF-IDF feature space followed by dimensionality reduction using autoencoder followed by OSVM for anomaly detec-

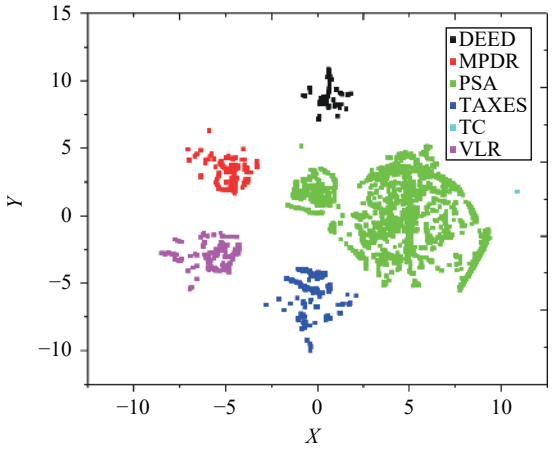

Fig. 4 Projection of documents using TF-IDF feature space on a 2D plane using t-SNE. Color versions of the figures in this paper are available online.

Table 1 Dimensions and sparsity of the document vectors represented using TF-IDF and Doc2Vec

	Dimension	Sparsity
TF-IDF	8 310 × 16 000	97.09%
Doc2Vec	8 310 × 300	0%

tion as shown in Fig. 5. The autoencoder model was trained separately with multiple latest dimensions. The generic architecture of both autoencoder training and the AD model is shown in Fig. 6.

TF-IDF (Term frequency – inverse document frequency): TF-IDF is a well-accepted technique[64] in the field of data mining, used for information extraction, search engines, the numerical representation of text data, etc. This is a metric of a combination of two numbers. This is a metric of a combination of two numbers: term frequency TF and inverse document frequency IDF. Term frequency $tf(t,d)$ is the frequency of a token within a document represented by $f_{t,d}$ where t is the term and d is the document. But just considering the frequency of a word within a document does not reflect the true importance of the word. The importance of a highly frequent word becomes less if that word is very frequent in all the documents of the corpus. The importance is proportional to the rarity of the word in the corpus.

Support vector machine: Two sets $X \in \{x_1, x_2, \cdots, x_n\}$ and $Y \in \{1, -1\}$ belonging to $\mathbf{R}^d$ are mapped by a surjective function $F : X \to Y$. Y here determines the class label of the data points[67]. SVM projects the data points in a higher dimension $\mathbf{R}^n$ where $n > d$ using a nonlinear function $\beth$. Data points which cannot be separated by a hyperplane of dimension d can now be separated by a hyperplane in dimension n.

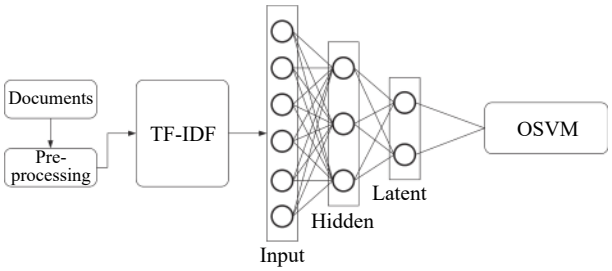

Fig. 5 Architecture 2: Engineered TF-IDF features using autoencoder used by OSVM for anomaly detection

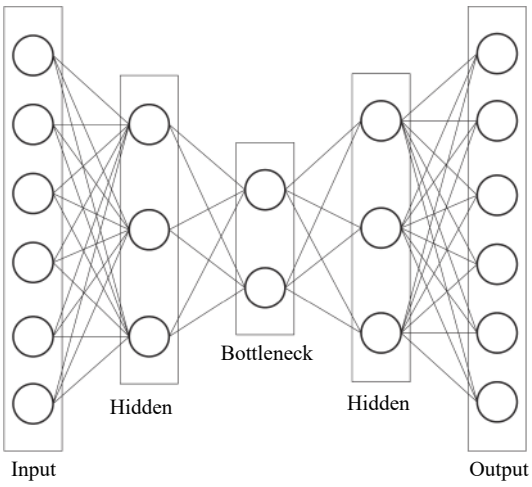

Fig. 6 Architecture 2: A generic architecture of an autoencoder

One-class SVM: One-class SVM continues the same philosophy of classification discussed above. There are two types of OSVM which has been extensively used interchangeably in different experiments over time. One is the hyperplane based maximum margin approach[20, 21] and the other takes a spherical boundary approach[21]. In the present study, the hyperplane based OSVM has been used.

In the first type, all the data points in the domain $\mathbf{R}^n$, are separated from the origin to place them as far as possible from the origin. The hyperplane which maximizes the distance of the data points from the origin is used for the binary function which returns $+1$ if the point is near the data points and -1 otherwise.

Another approach showed a variation with a hypersphere optimization instead of taking a hyper-planner approach[21]. In this approach, a sphere with center c and radius R, the volume R^2 is minimized to include all positive data points within the hypersphere. This approach is more effectively adopted for the anomaly detection scenarios.

Autoencoder: Not all the features in the TF-IDF feature space have equal importance in the representation of the document. It was necessary to extract the features with higher prominence in the documents and ignore the other features[68]. There were two advantages to this. Firstly, the dimensionality of the data was reduced and secondly, the important features were extracted for feature representation. Two well-known techniques of the dimensionality reduction are principal component analysis and autoencoder. The former is a linear dimensionality reduction technique that works on the variability of the component whereas AE is a nonlinear method of dimension reduction[22] that can find non-linear complex hidden patterns. In the present study, as we are dealing with vector representation of documents of a huge text corpus and the features have no linear relationship with each other, AE was a better choice of algorithm for this task.

AE network architecture for the present study: In the present study, a hybrid approach has been adopted to detect the anomalous documents. In the first stage, a neural network-based machine learning approach has been adopted for feature extraction and representation followed by OSVM for outlier detection. This architecture was tested with different input combinations to establish the finding. The overall architecture is described in Figs. 6–8.

Network layers: The input layer dimension in our study was 16 000 (obtained from TF-IDF representation) which we have represented in four different dimensions (32, 64, 128, 256) after reduction using autoencoder. The architecture of the network varied depending on the size of the bottleneck layer. Every hidden layer was designed to have a 50% reduction in terms of the previous layer dimension. For example, if the input layer had 16 000 di-

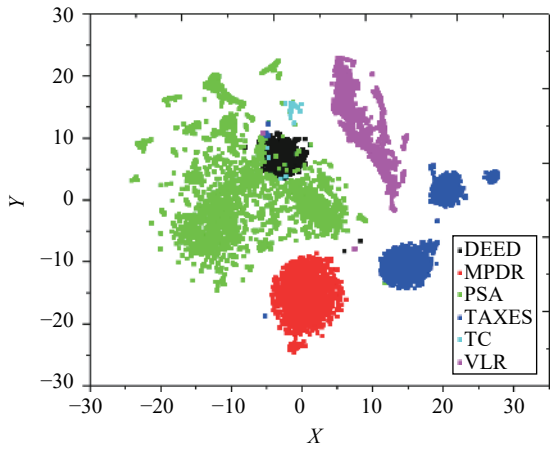

Fig. 7 Projection of documents using Doc2Vec embedding on a 2D plane using t-SNE

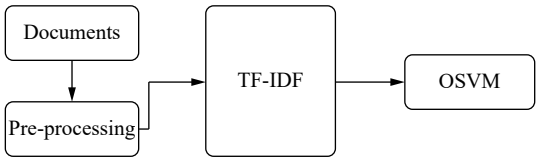

Fig. 8 Architecture 1: TF-IDF and Doc2Vec features directly used by O-SVM for anomaly detection

mensions (number of nodes), then the immediate next hidden layer had 8 000 nodes which is 50% of the size of the previous layer. Following this pattern, we have six-layered encoding for a target bottleneck of 256 dimensions, seven for 128 and eight for 64 and nine for 32.

Activation function: The choice of hyperparameters for the autoencoder has been completely based on the experiment. The activation function used for all the layers apart from the encoder side of the bottleneck is "tanh". The layer before the bottleneck has the activation function "elu". "tanh" being a hyperbolic tangent function produces output in the range of [−1, +1] whereas "elu" $f(x) = \begin{cases} x, & \text{if } x > 0, \\ a(\mathrm{e}^x - 1), & \text{if } x \leq 0 \end{cases}$ produces smooth range until the output reaches $-\alpha$. The range of possible values of TF-IDF vectors is always positive and the motive behind using a linear unit family activation function in the bottleneck layer is to allow the system to reproduce the values of the input representation.

Loss function: Cosine proximity (CP) was chosen as a preferred loss function over mean squared error (MSE) since the dataset used for the research is text. MSE is not an appropriate representation of similarity or dissimilarity among text data as it does not consider the angle between two vectors. It only considers the magnitude of vectors for calculating the dissimilarity where the similarity of content between two document vectors can be understood more by the cosine angle than the difference in magnitude. Given two vectors P and Q, the cosine similarity is represented as (4). The negative of cosine proximity is minimized in this loss function as part of the gradient descent.

$$\frac{P \cdot Q}{||P||\,||Q||} = \frac{\sum_{i=1}^{n} P_i Q_i}{\sqrt{\sum_{i=1}^{n} P_i^2}\sqrt{\sum_{i=1}^{n} Q_i^2}}. \tag{4}$$

4 Experiments

Two sets of experiments were conducted with the data to prepare a comparative study of the accuracy and performance of the anomaly detector. The data and processes were real-time. The experiment producing the best result in terms of both the accuracy and performance factors would be chosen for the implementation in a real-time ADMS. The uniqueness of the AD model here is that, multiple classes together form the positive class and the AD needs to detect all other documents as an anomaly. The experiment design is a real-time requirement from the industry.

Six document types were divided into two groups. In the first group, PSA, MPDR and TAXES document types were considered, and the other group had DEED, TC and VLR. Six combinations of document types for each group were studied based on the process definition of the title production system. There could be one more combination of positive classes possible for each group, i.e., the combination of all three document types. However, only the below combinations (Table 2) are considered for the present study.

Each combination was trained and tested with an eighty-twenty train and test split, taking both polynomial degree 3 and 4 and RBF kernels with ν value 0.05 to 0.4 incremented by 0.05 every time. Both inlier accuracy and outlier accuracy along with the training time were captured after every experiment.

In Experiments 2, the same combinations of docu-

Table 2 Combinations treated as positive class for the experiment

	Positive class
Group 1	PSA
	MPDR
	TAXES
	PSA+MPDR
	PSA+TAXES
	MPDR+TAXES
Group 2	DEED
	TC
	VLR
	DEED+TC
	DEED+VLR
	VLR+TC

ments were considered but the documents were represented in a lower-dimensional feature space using deep learning techniques. AE was used for the representation of the documents in latent space with much lower dimensions.

Experiment 1: The first set of experiments were conducted by directly sending the features into an OSVM model. The model was tuned with hyperparameters to obtain the best result. The training sample collected for the positive classes was manually verified and chosen and there was no possibility of the presence of outliers. The samples had variations, but the intent of the experiment was to make the model learn all possible variations of the samples. To find the optimal ν, it was varied from 0.01 to 0.5 to capture F1 score and Recall.

Observation: It is noticed that ν ranging from 0.05 to 0.01, the accuracy scores were stabilized (Fig. 9). Because of this reason, the ν value was kept at default 0.01 which meant that there was very little room for the presence of outliers. The value of γ was varied from the default which is $1/n \times var\,(x)$ to $\gamma \times 2^{10}$, increasing it by 2^{epoch} at every step.

3 014 positive samples were used for training, 754 positive and 2 603 negative samples were used for the testing. The experiment was conducted with two different kernels, RBF and polynomial with degrees 3 and 4. As an outcome, the positive class accuracy (PCA), F1 score, training time (TT) and average inference time (AIT) of the models were captured which is shown in Tables 3–14.

Observation: It is observed that the OSVM with RBF kernel performed steadily in terms of F1 score and PCA. The polynomial kernel with low γ value performed below expectation but the system performed much more steadily than that of RBF kernel when the γ value reached 0.5 and above.

Experiment 2: The second set of experiments was conducted with a reduced dimension of the TF-IDF feature space to obtain better performance. TF-IDF features were reduced to four lower dimensions (32, 64, 128 and 256) and considering the training and validation loss,

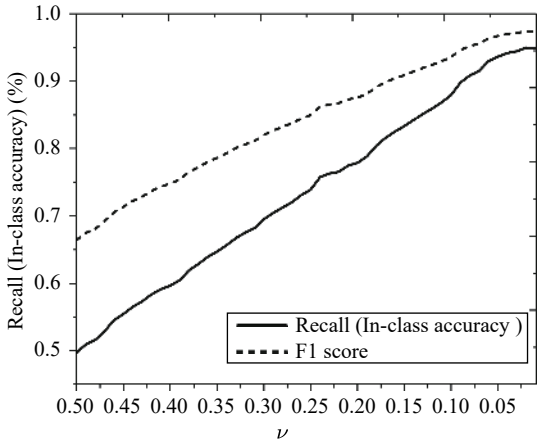

Fig. 9 Change of F1 and Recall with changing values of ν

a 64 dimension feature space was finalized as the input for the OSVM AD model. Reconstruction loss for both the training and validation sets were captured for the iterations and compared. Fig. 10 shows the loss convergence over the epochs for various dimensions. The reconstruction loss was calculated in terms of the cosine proximity whereas the model training was performed over MSE.

Observation: The validation loss of 64 dimensional latent space was the minimum during the convergence.

To confirm the above observation, four AE models with 32, 64, 128 and 256 dimensional latent space were trained only with the PSA document class and inferred with all document classes to calculate the actual reconstruction loss. If the AE model was able to train well, reconstruction loss of document types other than PSA should be high as the model did not learn to reconstruct those vectors. Fig. 11 plots capturing the reconstruction loss of different document types confirms that the assumption was true.

Observation: The median value of the reconstruction loss for 64 dimensional latent space was observed to be the minimum compared to the other three dimensions. Hence, the further experiments were conducted with the features represented in 64 dimensional latent space.

After this point, the same set of experiments as conducted in Experiment 1 were conducted in Experiment 2. All the combinations mentioned in Table 2 were considered as positive classes and the OSVM model was trained. In OSVM, both RBF and polynomial (degree 4) kernels were used along with varying υ values from 05 to 0.01. As we encountered acceptable accuracies with the default γ value of OSVM, i.e., "scale", the γ values were kept constant. For every experiment, the positive class accuracy and the F1 score were captured and compared. The results are shown in Figs. 12 and 13.

Observation: The hybrid approach of AD produces better results when the polynomial kernel is used. For seven positive class combinations, the accuracy was captured above 90% with the lower ν value. For two combinations, the optimal accuracy was obtained at two different ν levels. There are four combinations for which the F1 scores were not in an acceptable range. Further studies can be performed to improve the accuracies for these four combinations.

Performance: Apart from experimenting with accuracy, the aim of the study was to also compare the performance of the models. Training and inference timings of the models are the key factors in the real-time use in title production. Though the accuracies in the traditional models were well above 90% and in a business acceptable range and comparable with the hybrid approach, the reasoning behind dimensionality reduction was to make the model perform better without compromising the accuracy. Fig. 14 captures the training time and the average inference time of the predictions for both hybrid mod-

Table 3 Group 1 Combination 1, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
PSA	RBF	2	0.94619	0.8978	27.685	0.0085
		1	0.97554	0.9522	16.180	0.0046
		0.5	0.98108	0.9628	13.376	0.0033
		0.25	0.98314	0.9668	10.795	0.0027
		0.125	0.98655	0.9734	10.288	0.0024
		0.0625	0.98724	0.9748	9.7841	0.0022
		0.03125	0.98791	0.9761	9.3847	0.0022
		0.01562	0.98791	0.9761	9.3416	0.0023
		0.00781	0.98791	0.9761	8.9574	0.0021
		0.00390	0.98791	0.9761	8.5289	0.0018
		0.00195	0.98655	0.9734	7.7532	0.0017
		0.00097	0.98451	0.9694	7.0821	0.0014
		0.00048	0.97832	0.9575	5.9705	0.0010
		0.00024	0.99600	0.9920	5.2499	0.0009
		0.00012	0.99600	0.9920	4.6997	0.0009
		6.1E−05	0.43243	0.27582	3.88056	0.0009
	Poly	2	0.9151	0.8435	44.8306	0.0140
		1	0.9158	0.8448	44.1669	0.0138
		0.5	0.9135	0.8408	41.6490	0.0126
		0.25	0.2403	0.1366	13.9885	0.0042
		0.125	0.1173	0.0623	4.63337	0.0011
		0.0625	0.1243	0.0663	3.78379	0.0009
		0.03125	0.1243	0.0663	3.11586	0.0009
		0.01562	0.1243	0.0663	2.82930	0.0011
		0.00781	0.1243	0.0663	2.90496	0.0008
		0.00390	0.1243	0.0663	2.90937	0.0008
		0.00195	0.1243	0.0663	2.93636	0.0008
		0.00097	0.1243	0.0663	3.00021	0.0009
		0.00048	0.1243	0.0663	2.89418	0.0008
		0.00024	0.1243	0.0663	2.82138	0.0008
		0.00012	0.1243	0.0663	2.82865	0.0008
		6.1E−05	0.1243	0.0663	2.82616	0.0008

Table 4 Group 1 Combination 2, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
MPDR	RBF	2.0000	0.9506	0.9747	2.8080	0.0023
		1.0000	0.9658	0.9788	2.1094	0.0014
		0.5000	0.9620	0.9768	1.7351	0.0013
		0.2500	0.9696	0.9808	1.6913	0.0011
		0.1250	0.9696	0.9808	1.6164	0.0010
		0.0625	0.9734	0.9827	1.5424	0.0009
		0.0313	0.9696	0.9808	1.5254	0.0010
		0.0156	0.9696	0.9808	1.6757	0.0009
		0.0078	0.9696	0.9808	1.3725	0.0008
		0.0039	0.9658	0.9788	1.3124	0.0007
		0.0020	0.9696	0.9808	1.1665	0.0006
		0.0010	0.9506	0.9709	1.0167	0.0004
		0.0005	0.7148	0.8337	0.8053	0.0004
		0.0002	0.6654	0.7991	0.6781	0.0004
		0.0001	0.6768	0.8073	0.6210	0.0004
		6.1E−05	0.7300	0.8440	0.5765	0.0004
	Poly	2.0000	0.9392	0.9686	3.1121	0.0023
		1.0000	0.9468	0.9727	2.9506	0.0024
		0.5000	0.9430	0.9706	2.8929	0.0023
		0.2500	0.9049	0.9501	1.4003	0.0011
		0.1250	0.6350	0.7767	0.5627	0.0004
		0.0625	0.6996	0.8233	0.5998	0.0004
		0.0313	0.6996	0.8233	0.5067	0.0004
		0.0156	0.6996	0.8233	0.5039	0.0004
		0.0078	0.6996	0.8233	0.5406	0.0004
		0.0039	0.6996	0.8233	0.5482	0.0004
		0.0020	0.6996	0.8233	0.5186	0.0004
		0.0010	0.6996	0.8233	0.5150	0.0004
		0.0005	0.6996	0.8233	0.5280	0.0004
		0.0002	0.6996	0.8233	0.5109	0.0004
		0.0001	0.6996	0.8233	0.5154	0.0004
		6.1E−05	0.6996	0.8233	0.5033	0.0004

Table 5 Group 1 Combination 3, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
TAXES	RBF	2.0000	0.9729	0.9824	2.7063	0.0022
		1.0000	0.9767	0.9844	2.1335	0.0017
		0.5000	0.9845	0.9883	1.6854	0.0012
		0.2500	0.9806	0.9864	1.4338	0.0010
		0.1250	0.9806	0.9864	1.4737	0.0010
		0.0625	0.9806	0.9864	1.4375	0.0009
		0.0313	0.9806	0.9864	1.5722	0.0010
		0.0156	0.9806	0.9864	1.4561	0.0009
		0.0078	0.9806	0.9864	1.3805	0.0009
		0.0039	0.9806	0.9864	1.2332	0.0007
		0.0020	0.9535	0.9723	1.1323	0.0007
		0.0010	0.6822	0.8073	1.1180	0.0007
		0.0005	0.9496	0.9703	0.8654	0.0005
		0.0002	0.3372	0.5014	0.7411	0.0004
		0.0001	0.0194	0.0380	0.5813	0.0004
		6.1E−05	0.0271	0.0528	0.5506	0.0004
	Poly	2.0000	0.9651	0.9822	2.6744	0.0022
		1.0000	0.9612	0.9802	2.6948	0.0022
		0.5000	0.9574	0.9782	2.5096	0.0020
		0.2500	0.8721	0.9317	1.6069	0.0013
		0.1250	0.0194	0.0380	0.5342	0.0004
		0.0625	0.0194	0.0380	0.5742	0.0004
		0.0313	0.0194	0.0380	0.5205	0.0004
		0.0156	0.0194	0.0380	0.5363	0.0004
		0.0078	0.0194	0.0380	0.4988	0.0004
		0.0039	0.0194	0.0380	0.5705	0.0004
		0.0020	0.0194	0.0380	0.5080	0.0004
		0.0010	0.0194	0.0380	0.5038	0.0004
		0.0005	0.0194	0.0380	0.5116	0.0004
		0.0002	0.0194	0.0380	0.5093	0.0004
		0.0001	0.0194	0.0380	0.5266	0.0004
		6.1E−05	0.0194	0.0380	0.5051	0.0004

Table 6 Group 1 Combination 4, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
PSA+MPDR	RBF	2.0000	0.9145	0.9553	56.3283	0.0125
		1.0000	0.9607	0.9799	33.3234	0.0069
		0.5000	0.9744	0.9871	27.0313	0.0052
		0.2500	0.9794	0.9896	22.2409	0.0043
		0.1250	0.9803	0.9901	20.6766	0.0037
		0.0625	0.9833	0.9916	19.8318	0.0035
		0.0313	0.9823	0.9911	20.2884	0.0034
		0.0156	0.9823	0.9911	19.6910	0.0033
		0.0078	0.9833	0.9916	20.9988	0.0032
		0.0039	0.9813	0.9906	18.2908	0.0030
		0.0020	0.9803	0.9901	16.1793	0.0025
		0.0010	0.9626	0.9810	15.0767	0.0022
		0.0005	0.9312	0.9644	12.5976	0.0018
		0.0002	0.9794	0.9896	11.8095	0.0015
		0.0001	0.5556	0.7143	9.8810	0.0015
		6.1E−05	0.5329	0.6953	8.4059	0.0016
	Poly	2.0000	0.8692	0.9300	83.7114	0.0195
		1.0000	0.8692	0.9300	83.9049	0.0198
		0.5000	0.8663	0.9283	76.4466	0.0177
		0.2500	0.2507	0.4009	30.9581	0.0069
		0.1250	0.6175	0.7635	6.9060	0.0015
		0.0625	0.6962	0.8209	6.3406	0.0015
		0.0313	0.6962	0.8209	6.4808	0.0016
		0.0156	0.6962	0.8209	6.7475	0.0016
		0.0078	0.6962	0.8209	6.4375	0.0015
		0.0039	0.6962	0.8209	6.6395	0.0015
		0.0020	0.6962	0.8209	6.4622	0.0015
		0.0010	0.6962	0.8209	6.4961	0.0015
		0.0005	0.6962	0.8209	6.4113	0.0015
		0.0002	0.6962	0.8209	6.7328	0.0016
		0.0001	0.6962	0.8209	6.3336	0.0015
		6.1E−05	0.6962	0.8209	6.3235	0.0016

Table 7 Group 1 Combination 5, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
PSA+TAXES	RBF	2.0000	0.9051	0.9502	54.9255	0.0124
		1.0000	0.9407	0.9695	32.3525	0.0064
		0.5000	0.9565	0.9778	24.5071	0.0046
		0.2500	0.9704	0.9850	21.9204	0.0039
		0.1250	0.9743	0.9870	20.4208	0.0035
		0.0625	0.9733	0.9865	19.8260	0.0034
		0.0313	0.9773	0.9885	19.5137	0.0033
		0.0156	0.9773	0.9885	19.0924	0.0034
		0.0078	0.9763	0.9880	18.8371	0.0035
		0.0039	0.9713	0.9855	18.2730	0.0030
		0.0020	0.9664	0.9829	16.2048	0.0025
		0.0010	0.9713	0.9855	14.2056	0.0021
		0.0005	0.7945	0.8855	12.2956	0.0017
		0.0002	0.9931	0.9965	11.8141	0.0015
		0.0001	0.9960	0.9980	10.4107	0.0015
		6.1E−05	0.9901	0.9950	8.5068	0.0016
	Poly	2.0000	0.8626	0.9263	87.3693	0.0202
		1.0000	0.8636	0.9268	84.9025	0.0192
		0.5000	0.8557	0.9223	78.2043	0.0179
		0.2500	0.1591	0.2745	30.8067	0.0068
		0.1250	0.1285	0.2277	7.2576	0.0016
		0.0625	0.2964	0.4573	6.4696	0.0015
		0.0313	0.2964	0.4573	6.3681	0.0015
		0.0156	0.2964	0.4573	6.4391	0.0015
		0.0078	0.2964	0.4573	6.4442	0.0015
		0.0039	0.2964	0.4573	6.4331	0.0015
		0.0020	0.2964	0.4573	6.3368	0.0015
		0.0010	0.2964	0.4573	6.5316	0.0015
		0.0005	0.2964	0.4573	6.4667	0.0015
		0.0002	0.2964	0.4573	6.3241	0.0015
		0.0001	0.2964	0.4573	6.3171	0.0015
		6.1E−05	0.2964	0.4573	6.5906	0.0015

Table 8 Group 1 Combination 6, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
MPDR+TAXES	RBF	2	0.9367	0.9654	8.9578	0.0042
		1	0.9635	0.9795	7.5232	0.0029
		0.5	0.9770	0.9864	6.0763	0.0022
		0.25	0.9808	0.9884	5.6920	0.0019
		0.125	0.9846	0.9903	4.8941	0.0019
		0.0625	0.9846	0.9903	5.2747	0.0018
		0.0313	0.9846	0.9903	5.4708	0.0017
		0.0156	0.9827	0.9894	5.1279	0.0017
		0.0078	0.9827	0.9894	5.1108	0.0016
		0.0039	0.9770	0.9864	3.9053	0.0015
		0.0020	0.9578	0.9765	3.8955	0.0012
		0.0010	0.9290	0.9613	3.5238	0.0010
		0.0005	0.8868	0.9381	3.2836	0.0009
		0.0002	0.9962	0.9962	2.8364	0.0008
		0.0001	0.9789	0.9874	2.4240	0.0008
		6.1E−05	0.0403	0.0772	1.9183	0.0008
	Poly	2.0000	0.9443	0.9714	10.5177	0.0045
		1.0000	0.9424	0.9704	10.5182	0.0044
		0.5000	0.9463	0.9724	10.0883	0.0042
		0.2500	0.8100	0.8950	5.7079	0.0022
		0.1250	0.0787	0.1459	2.1876	0.0008
		0.0625	0.0480	0.0916	1.8155	0.0008
		0.0313	0.0480	0.0916	1.7957	0.0008
		0.0156	0.0480	0.0916	1.7299	0.0008
		0.0078	0.0480	0.0916	1.8478	0.0009
		0.0039	0.0480	0.0916	1.6383	0.0007
		0.0020	0.0480	0.0916	1.9589	0.0008
		0.0010	0.0480	0.0916	1.7586	0.0008
		0.0005	0.0480	0.0916	1.8024	0.0008
		0.0002	0.0480	0.0916	1.7519	0.0008
		0.0001	0.0480	0.0916	1.8995	0.0008
		6.1E−05	0.0480	0.0916	1.7568	0.0007

Table 9 Group 2 Combination 1, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
DEED	RBF	2.0000	0.2963	0.4571	4.1625	0.0105
		1.0000	0.5833	0.7368	2.2738	0.0057
		0.5000	0.7222	0.8254	1.4766	0.0036
		0.2500	0.7407	0.8290	1.1915	0.0030
		0.1250	0.7593	0.8325	1.1533	0.0027
		0.0625	0.7685	0.8384	1.1606	0.0025
		0.0313	0.7778	0.8442	0.9847	0.0022
		0.0156	0.7500	0.8265	0.8861	0.0019
		0.0078	0.7130	0.8021	0.6885	0.0013
		0.0039	0.5556	0.7018	0.5796	0.0010
		0.0020	0.1667	0.2857	0.3696	0.0006
		0.0010	0.0000	0.0000	0.1692	0.0003
		0.0005	0.1204	0.2149	0.1260	0.0002
		0.0002	0.0833	0.1538	0.1032	0.0002
		0.0001	0.0833	0.1538	0.1039	0.0002
		6.1E−05	0.0833	0.1538	0.1057	0.0002
	Poly	2.0000	0.1759	0.2992	6.0433	0.0125
		1.0000	0.1759	0.2992	5.4637	0.0095
		0.5000	0.1296	0.2295	4.0447	0.0095
		0.2500	0.0000	0.0000	0.3503	0.0005
		0.1250	0.0000	0.0000	0.0925	0.0002
		0.0625	0.0000	0.0000	0.1025	0.0002
		0.0313	0.0000	0.0000	0.0975	0.0002
		0.0156	0.0000	0.0000	0.0947	0.0002
		0.0078	0.0000	0.0000	0.0930	0.0002
		0.0039	0.0000	0.0000	0.0907	0.0002
		0.0020	0.0000	0.0000	0.0918	0.0002
		0.0010	0.0000	0.0000	0.0990	0.0002
		0.0005	0.0000	0.0000	0.0929	0.0002
		0.0002	0.0000	0.0000	0.0907	0.0002
		0.0001	0.0000	0.0000	0.0960	0.0002
		6.1E−05	0.0000	0.0000	0.0929	0.0002

Table 10 Group 2 Combination 2, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
TC	RBF	2.0000	0.9145	0.9553	56.3283	0.0125
		1.0000	0.9607	0.9799	33.3234	0.0069
		0.5000	0.9744	0.9871	27.0313	0.0052
		0.2500	0.9794	0.9896	22.2409	0.0043
		0.1250	0.9803	0.9901	20.6766	0.0037
		0.0625	0.9833	0.9916	19.8318	0.0035
		0.0313	0.9823	0.9911	20.2884	0.0034
		0.0156	0.9823	0.9911	19.6910	0.0033
		0.0078	0.9833	0.9916	20.9988	0.0032
		0.0039	0.9813	0.9906	18.2908	0.0030
		0.0020	0.9803	0.9901	16.1793	0.0025
		0.0010	0.9626	0.9810	15.0767	0.0022
		0.0005	0.9312	0.9644	12.5976	0.0018
		0.0002	0.9794	0.9896	11.8095	0.0015
		0.0001	0.5556	0.7143	9.8810	0.0015
		6.1E−05	0.5329	0.6953	8.4059	0.0016
	Poly	2.0000	0.8692	0.9300	83.7114	0.0195
		1.0000	0.8692	0.9300	83.9049	0.0198
		0.5000	0.8663	0.9283	76.4466	0.0177
		0.2500	0.2507	0.4009	30.9581	0.0069
		0.1250	0.6175	0.7635	6.9060	0.0015
		0.0625	0.6962	0.8209	6.3406	0.0015
		0.0313	0.6962	0.8209	6.4808	0.0016
		0.0156	0.6962	0.8209	6.7475	0.0016
		0.0078	0.6962	0.8209	6.4375	0.0015
		0.0039	0.6962	0.8209	6.6395	0.0015
		0.0020	0.6962	0.8209	6.4622	0.0015
		0.0010	0.6962	0.8209	6.4961	0.0015
		0.0005	0.6962	0.8209	6.4113	0.0015
		0.0002	0.6962	0.8209	6.7328	0.0016
		0.0001	0.6962	0.8209	6.3336	0.0015
		6.1E−05	0.6962	0.8209	6.3235	0.0016

Table 11 Group 2 Combination 3, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
VLR	RBF	2.0000	0.9580	0.9786	1.4337	0.0010
		1.0000	0.9771	0.9884	1.0730	0.0007
		0.5000	0.9771	0.9884	1.0679	0.0006
		0.2500	0.9809	0.9904	0.9643	0.0006
		0.1250	0.9809	0.9904	0.8669	0.0005
		0.0625	0.9847	0.9923	0.8693	0.0005
		0.0313	0.9847	0.9923	0.8648	0.0005
		0.0156	0.9847	0.9923	0.8685	0.0006
		0.0078	0.9847	0.9923	0.8288	0.0006
		0.0039	0.9847	0.9923	0.9024	0.0005
		0.0020	0.9809	0.9904	0.8109	0.0005
		0.0010	0.9656	0.9825	0.7909	0.0005
		0.0005	0.6870	0.8145	0.6472	0.0004
		0.0002	0.1641	0.2820	0.5807	0.0003
		0.0001	0.1221	0.2177	0.5125	0.0003
		6.1E−05	0.0420	0.0806	0.4612	0.0003
	Poly	2.0000	0.9542	0.9766	1.2889	0.0010
		1.0000	0.9542	0.9766	1.2768	0.0010
		0.5000	0.9466	0.9725	1.2818	0.0010
		0.2500	0.9237	0.9603	1.1165	0.0008
		0.1250	0.2061	0.3418	0.4875	0.0003
		0.0625	0.0382	0.0735	0.4312	0.0004
		0.0313	0.0382	0.0735	0.4514	0.0003
		0.0156	0.0382	0.0735	0.3949	0.0003
		0.0078	0.0382	0.0735	0.4018	0.0003
		0.0039	0.0382	0.0735	0.4073	0.0003
		0.0020	0.0382	0.0735	0.4341	0.0004
		0.0010	0.0382	0.0735	0.4572	0.0004
		0.0005	0.0382	0.0735	0.4584	0.0004
		0.0002	0.0382	0.0735	0.4171	0.0003
		0.0001	0.0382	0.0735	0.3956	0.0003
		6.1E−05	0.0382	0.0735	0.4030	0.0003

Table 12 Group 2 Combination 4, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
TC+DEED	RBF	2.0000	0.2960	0.4568	5.0269	0.0113
		1.0000	0.6320	0.7745	2.8451	0.0061
		0.5000	0.7520	0.8584	2.0213	0.0040
		0.2500	0.8160	0.8987	1.6556	0.0035
		0.1250	0.8480	0.9177	1.5253	0.0033
		0.0625	0.8560	0.9224	1.4713	0.0030
		0.0313	0.8640	0.9270	1.2338	0.0025
		0.0156	0.8400	0.9130	1.0961	0.0021
		0.0078	0.7520	0.8584	0.8491	0.0013
		0.0039	0.6320	0.7745	0.6452	0.0009
		0.0020	0.1520	0.2639	0.3289	0.0006
		0.0010	0.0240	0.0469	0.2878	0.0005
		0.0005	0.0000	0.0000	0.1932	0.0003
		0.0002	0.0000	0.0000	0.1258	0.0002
		0.0001	0.0000	0.0000	0.1266	0.0002
		6.1E−05	0.0000	0.0000	0.1268	0.0002
	Poly	2.0000	0.1760	0.2993	8.2907	0.0139
		1.0000	0.1680	0.2877	8.2532	0.0139
		0.5000	0.1280	0.2270	5.1765	0.0109
		0.2500	0.0000	0.0000	0.4287	0.0006
		0.1250	0.0000	0.0000	0.1465	0.0002
		0.0625	0.0000	0.0000	0.1130	0.0002
		0.0313	0.0000	0.0000	0.1064	0.0002
		0.0156	0.0000	0.0000	0.1346	0.0002
		0.0078	0.0000	0.0000	0.1198	0.0002
		0.0039	0.0000	0.0000	0.1081	0.0002
		0.0020	0.0000	0.0000	0.1080	0.0002
		0.0010	0.0000	0.0000	0.1047	0.0002
		0.0005	0.0000	0.0000	0.1077	0.0002
		0.0002	0.0000	0.0000	0.1164	0.0002
		0.0001	0.0000	0.0000	0.1097	0.0002
		6.1E−05	0.0000	0.0000	0.1083	0.0002

Table 13 Group 2 Combination 5, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
DEED+VLR	RBF	2.0000	0.8027	0.8906	13.8363	0.0099
		1.0000	0.8730	0.9322	8.5434	0.0052
		0.5000	0.9243	0.9580	5.7540	0.0037
		0.2500	0.9297	0.9542	5.6717	0.0035
		0.1250	0.9378	0.9572	5.4588	0.0027
		0.0625	0.9432	0.9601	5.3236	0.0027
		0.0313	0.9459	0.9602	5.1715	0.0027
		0.0156	0.9514	0.9631	4.9427	0.0025
		0.0078	0.9405	0.9587	4.4087	0.0022
		0.0039	0.9243	0.9474	3.3093	0.0017
		0.0020	0.8568	0.9135	2.6842	0.0014
		0.0010	0.7135	0.8328	1.8711	0.0011
		0.0005	0.3568	0.5217	1.2479	0.0005
		0.0002	0.3378	0.5051	1.0675	0.0004
		0.0001	0.5649	0.7219	0.9909	0.0006
		6.1E−05	0.3838	0.5547	0.9499	0.0005
	Poly	2.0000	0.7703	0.8702	21.4893	0.0135
		1.0000	0.7676	0.8685	20.5118	0.0130
		0.5000	0.7432	0.8527	19.2422	0.0122
		0.2500	0.0027	0.0054	3.8471	0.0021
		0.1250	0.3676	0.5375	0.9317	0.0005
		0.0625	0.3622	0.5317	0.7855	0.0005
		0.0313	0.3622	0.5317	0.7291	0.0004
		0.0156	0.3622	0.5317	0.7335	0.0004
		0.0078	0.3622	0.5317	0.7313	0.0004
		0.0039	0.3622	0.5317	0.7230	0.0005
		0.0020	0.3622	0.5317	0.7273	0.0004
		0.0010	0.3622	0.5317	0.7410	0.0004
		0.0005	0.3622	0.5317	0.7507	0.0004
		0.0002	0.3622	0.5317	0.7421	0.0004
		0.0001	0.3622	0.5317	0.7319	0.0005
		6.1E−05	0.3622	0.5317	0.7663	0.0005

Table 14 Group 2 Combination 6, Kernel: RBF and polynomial

PC	K	γ	F1	PCA	TT	AIT
VLR+TC	RBF	2.0000	0.9173	0.9568	2.7324	0.0021
		1.0000	0.9424	0.9704	2.1166	0.0014
		0.5000	0.9676	0.9835	1.4550	0.0010
		0.2500	0.9676	0.9835	1.2487	0.0008
		0.1250	0.9676	0.9835	1.2957	0.0008
		0.0625	0.9676	0.9835	1.2950	0.0008
		0.0313	0.9676	0.9835	1.3703	0.0010
		0.0156	0.9712	0.9854	1.2851	0.0008
		0.0078	0.9712	0.9854	1.2289	0.0007
		0.0039	0.9676	0.9835	1.2312	0.0008
		0.0020	0.9532	0.9761	1.0896	0.0006
		0.0010	0.9532	0.9761	0.8541	0.0004
		0.0005	0.7626	0.8653	0.7895	0.0004
		0.0002	0.6007	0.7506	0.7503	0.0004
		0.0001	0.6619	0.7965	0.6940	0.0004
		6.1E−05	0.6295	0.7726	0.5639	0.0004
	Poly	2.0000	0.9101	0.9529	3.0334	0.0022
		1.0000	0.9101	0.9529	3.3766	0.0027
		0.5000	0.9137	0.9549	3.1853	0.0022
		0.2500	0.6655	0.7991	1.9250	0.0013
		0.1250	0.5791	0.7335	0.6362	0.0004
		0.0625	0.4856	0.6538	0.4570	0.0003
		0.0313	0.4856	0.6538	0.4519	0.0004
		0.0156	0.4856	0.6538	0.4538	0.0004
		0.0078	0.4856	0.6538	0.4671	0.0004
		0.0039	0.4856	0.6538	0.4553	0.0003
		0.0020	0.4856	0.6538	0.4738	0.0003
		0.0010	0.4856	0.6538	0.4774	0.0004
		0.0005	0.4856	0.6538	0.4531	0.0004
		0.0002	0.4856	0.6538	0.4729	0.0004
		0.0001	0.4856	0.6538	0.4903	0.0004
		6.1E−05	0.4856	0.6538	0.5049	0.0004

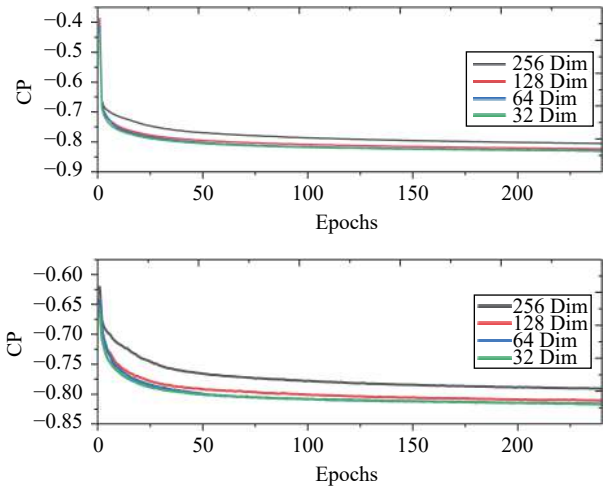

Fig. 10 Autoencoder training and validation loss over the epochs

els as well as the models without the AE.

Observation: The hybrid model has been observed to have performed much better and consistently, concerning both training and inference time, than that of traditional OSVM.

5 Results and discussions

This section evaluates the results obtained through different experiments conducted using the proposed hybrid AD model. The experiments intend to measure the metrics as follows:

1) Evaluate the effect of kernel choice on the prediction accuracy for OSVM.

2) Compare the accuracy measures of using different kernel methods as well as different values of ν and γ in the experiments using OSVM.

3) Determine the effect in training time with and without using autoencoder based feature extractors and dimensionality reduction techniques.

4) Determine the optimal dimension of the feature space by experimenting with multiple sized latent space vectors as input to the OSVM.

The accuracy of the model is determined by the principle stated below:

Accuracy: The principle of evaluation of AD is like a binary classifier. There is a two-class prediction involved in anomaly detection. One class is positive ($+ve$) or normal and the other is negative or anomalous ($-ve$). There are four outcomes of the prediction. **True positive (TP)**: predictions that are correctly identified as an anomaly, **False positive (FP):** predictions that wrongly predicted abnormal data points as normal, **True negative (TN)**: predictions that correctly identified the anomalous data points, **False negative (FN):** predictions that incorrectly predicted normal data as an anomaly. Four basic measures of the predictions, *Errorrate*, *Accuracy*, *Sensitivity*, and *Specificity* are calculated from the confusion matrix (Table 15) generated from the prediction where the total number of positive class samples is represented by P and the total number of negative class samples is represented by N.

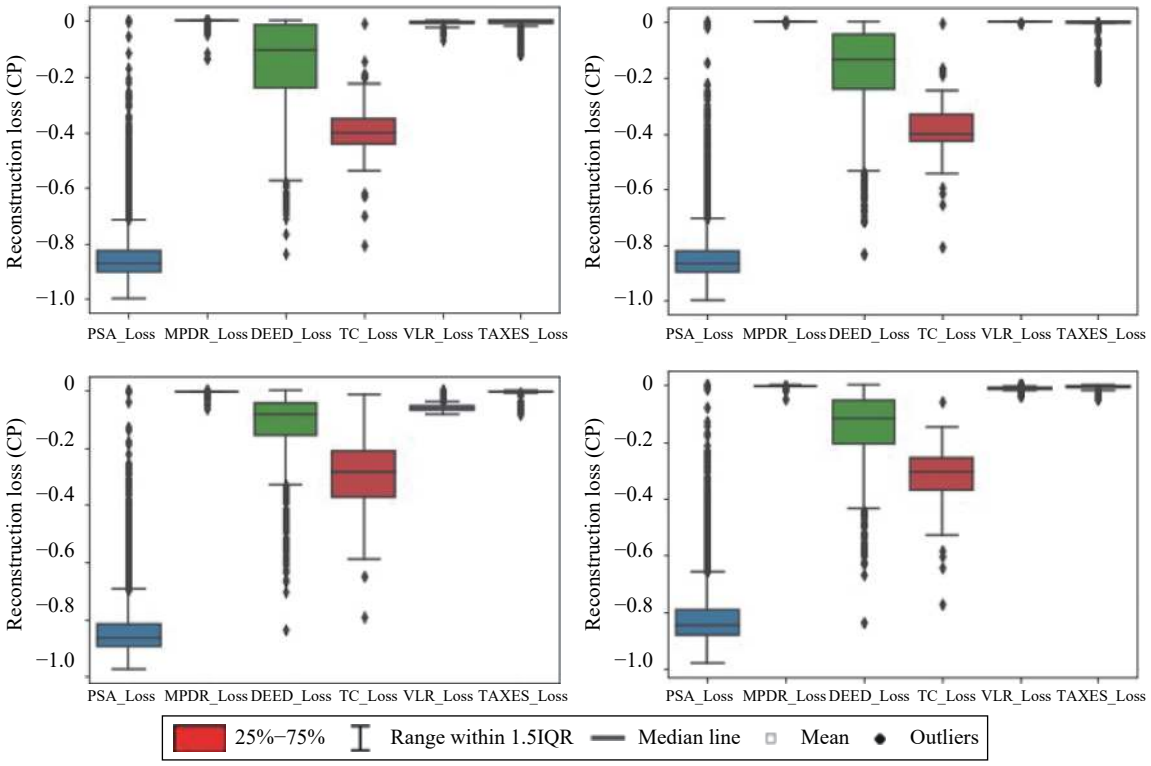

Fig. 11 Reconstruction loss distribution grouped by classes

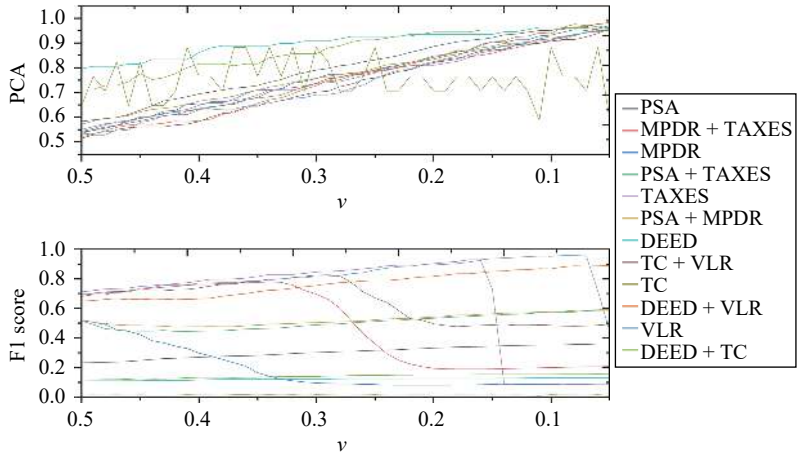

Fig. 12 PCA and F1-score with respect to v. Kernel function: RBF

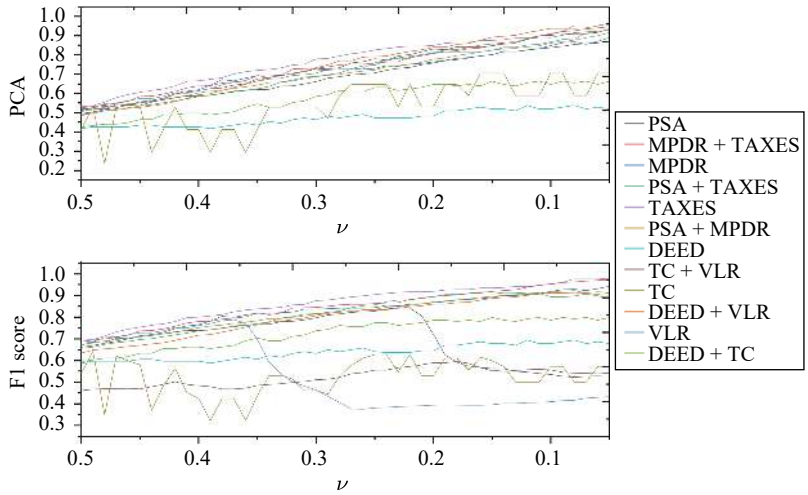

Fig. 13 PCA and F1 score with respect to ν. Kernel function: polynomial.

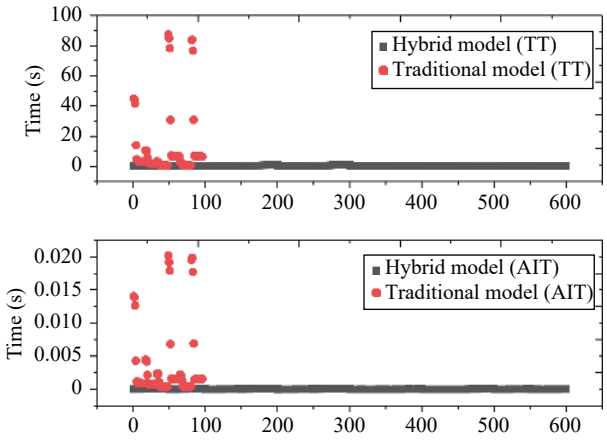

Fig. 14 Training time (TT) and Average inference time (AIT) comparison for hybrid and traditional model

$$Error\ rate = \frac{(FP + FN)}{P + N} \tag{5}$$

$$Accuracy = \frac{(TP + TN)}{P + N} \tag{6}$$

$$Sensitivity = \frac{TP}{P} \tag{7}$$

$$Specificity = \frac{TN}{N} \tag{8}$$

$$F1 = \frac{2Precision \times Recall}{Precision + Recall}. \tag{9}$$

Effect of kernel choice: The choice of the kernel has a huge impact on the performance of OSVM. In both the experiments, we used both RBF and polynomial kernels and gathered the results. In Experiment 1, it was noticed that the performance of models using RBF kernels in terms of both accuracy and timing, is better than that of the Polynomial kernel (Tables 4–15).

However, Experiment 2, the polynomial kernels outperformed the RBF kernel-based model for most of the combinations of positive classes. The F1 scores of different models based on the polynomial kernel showed a lower range of accuracies and inconsistencies (Fig. 12 and 13).

Effect of ν and γ: In Experiment 1, the optimal ν value was obtained between 0.01 and 0.05 from experi-

Table 15 Confusion matrix

		Predicted	
		Positive	Negative
Observed	Positive	TP	FN
	Negative	FP	TN

menting with different ν values within a range of 0.01 to 0.5 (Fig. 9). Keeping the ν value constant, Experiment 1 was conducted varying γ from 0.0000625 to 2. For different combinations of positive classes, different optimal γ were obtained. In Experiment 2, the γ value was kept constant at the default and the optimal ν value was obtained for different combinations.

Effect of hybrid technique in training and inference time: The effect of dimension on the training time and inference time has been captured in Fig. 14, where it is visible that the hybrid model is a clear winner in terms of the training and the inference time. Both the times have a steady performance whereas the traditional model with high dimensional data produces different timings for different models which are on the much higher end.

In summary, from the analysis of the experimental results, it is observed that the proposed hybrid approach of anomaly detection for high dimensional text data produces comparable accuracy (both PCA and F1 score) which are well within the real-time business acceptable range. In addition, the hybrid approach indubitably improves model performance in terms of training and inference. Considering the variety of data in terms of document types and the positive class combinations which were tested in this study, a generalization of this approach towards real-time ADMS is demonstrated.

6 Conclusions

In this paper, we presented a hybrid approach of AD by combining a traditional one-class classification algorithm called OSVM and a deep learning-based, self-supervised, non-linear dimensionality reduction algorithm named autoencoder. The novelty of the study lies in the choice of real-time problem and high dimensional text data of title insurance. Also, unlike other anomaly detection studies, we not only used data of one specific class as a normal or positive class sample but combinations of multiple classes as the normal or positive class. This increases the complexity of the problem because the AD model learns the distribution of multiple populations together considering those as a single population. As the study is influenced and motivated by a real-time business problem, the performance of the overall AD system is an important parameter of evaluation. The hybrid model matches the accuracy of the traditional approach and exceeds the performance in terms of both training and inference timing.

Though the study was performed with the data from the domain of TI, it does not limit the applicability to this domain only. The necessity of AD, in a composite positive class scenario in other domains handling image documents is ubiquitous in many other business scenarios of the present day.

One of the limitations of the proposed hybrid approach is that there is a need for two-fold training. One is for the autoencoder and the other is for the OSVM. Converting the hybrid architecture into a single training deep learning-based model with an objective of concept learning by injecting appropriate cost function would improve the usability and maintainability of such architecture[16, 69]. Secondly, the training and testing data were randomly sampled from a data lake where there is a possibility of missing out certain variations. A generative approach of training the autoencoder would improve the possibility of unseen sample generation which could improve the possibility of better performance of unseen data of similar classes. Presently, we are working on addressing both the improvements.

References

[1] X. D. Xu, H. W. Liu, M. H. Yao. Recent progress of anomaly detection. *Complexity*, vol. 2019, Article number 2686378, 2019. DOI: 10.1155/2019/2686378.

[2] Y. Hao, Z. J. Xu, Y. Liu, J. Wang, J. L. Fan. Effective crowd anomaly detection through spatio-temporal texture analysis. *International Journal of Automation and Computing*, vol. 16, no. 1, pp. 27–39, 2019. DOI: 10.1007/s11633-018-1141-z.

[3] M. Anderka, B. Stein, N. Lipka. Detection of text quality flaws as a one-class classification problem. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ACM, Glasgow, UK, pp. 2313–2316, 2011. DOI: 10.1145/2063576.2063954.

[4] Z. G. Ding, D. J. Du, M. R. Fei. An isolation principle based distributed anomaly detection method in wireless sensor networks. *International Journal of Automation and Computing*, vol. 12, no. 4, pp. 402–412, 2015. DOI: 10.1007/s11633-014-0847-9.

[5] V. Chandola, A. Banerjee, V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, vol. 41, no. 3, Article number 15, 2009. DOI: 10.1145/1541880.1541882.

[6] S. S. Khan, M. G. Madden. One-class classification: Taxonomy of study and review of techniques. *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014. DOI: 10.1017/S026988891300043X.

[7] M. Kemmler, E. Rodner, E. S. Wacker, J. Denzler. One-class classification with Gaussian processes. *Pattern Recognition*, vol. 46, no. 12, pp. 3507–3518, 2013. DOI: 10.1016/j.patcog.2013.06.005.

[8] Q. Leng, H. G. Qi, J. Miao, W. T. Zhu, G. P. Su. One-class classification with extreme learning machine. *Mathematical Problems in Engineering*, vol. 2015, Article number 412957, 2015. DOI: 10.1155/2015/412957.

[9] P. F. Liang, W. T. Li, H. Tian, J. L. Hu. One-class classification using a support vector machine with a quasi-linear kernel. *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, no. 3, pp. 449–456, 2019. DOI: 10.1002/tee.22826.

[10] C. Bellinger, S. Sharma, N. Japkowicz. One-class versus binary classification: Which and when? In *Proceedings of the 11th International Conference on Machine Learning and Applications*, IEEE, Boca Raton, USA, pp.102–106, 2012. DOI: 10.1109/ICMLA.2012.212.

[11] A. Guha, D. Samanta. Real-time application of document classification based on machine learning. In *Proceedings of the 1st International Conference on Information, Communication and Computing Technology*, Springer, Istanbul, Turkey, pp.366–379, 2020. DOI: 10.1007/978-3-030-38501-9_37.

[12] Y. Chen, M. J. Zaki. Kate: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Halifax, Canada, pp.85–94, 2017. DOI: 10.1145/3097983.3098017.

[13] D. Cozzolino, L. Verdoliva. Single-image splicing localization through autoencoder-based anomaly detection. In *Proceedings of IEEE International Workshop on Information Forensics and Security*, IEEE, Abu Dhabi, United Arab Emirates, 2016. DOI: 10.1109/WIFS.2016.7823921.

[14] D. Y. Oh, I. D. Yun. Residual error based anomaly detection using auto-encoder in SMD machine sound. *Sensors*, vol. 18, Article number 1308, 2018. DOI: 10.3390/s18051308.

[15] J. Mourao-Miranda, D. R. Hardoon, T. Hahn, A. F. Marquand, S. C. R. Williams, J. Shawe-Taylor, M. Brammer. Patient classification as an outlier detection problem: An application of the one-class support vector machine. *NeuroImage*, vol. 58, no. 3, pp. 793–804, 2011. DOI: 10.1016/j.neuroimage.2011.06.042.

[16] L. M. Manevitz, M. Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 139–154, 2001.

[17] T. Sukchotrat, S. B. Kim, F. Tsung. One-class classification-based control charts for multivariate process monitoring. *IIE Transactions*, vol. 42, no. 2, pp. 107–120, 2009. DOI: 10.1080/07408170903019150.

[18] P. Perera, V. M. Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5450–5463, 2019. DOI: 10.1109/TIP.2019.2917862.

[19] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Muller, M. Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp.4393–4402, 2018.

[20] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, ACM, Denver, USA, pp.582–588, 1999.

[21] D. M. J. Tax, R. P. W. Duin. Support vector data description. *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004. DOI: 10.1023/B:MACH.0000008084.60811.49.

[22] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*, Cambridge, USA: MIT Press, 2016.

[23] M. Sakurada, T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2nd Workshop on Machine Learning for Sensory Data Analysis*, ACM, Gold Coast, Australia, pp.4–11, 2014. DOI: 10.1145/2689746.2689747.

[24] M. Goldstein, S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS One*, vol. 11, no. 4, Article number e0152173, 2016. DOI: 10.1371/journal.pone.0152173.

[25] S. S. Khan, M. G. Madden. A survey of recent trends in one class classification. In *Proceedings of the 20th Irish Conference on Artificial Intelligence and Cognitive Science*, Springer, Dublin, Ireland, pp.188–197, 2010. DOI: 10.1007/978-3-642-17080-5_21.

[26] V. Mahadevan, W. X. Li, V. Bhalodia, N. Vasconcelos. Anomaly detection in crowded scenes. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Francisco, USA, pp.1975–1981, 2010. DOI: 10.1109/CVPR.2010.5539872.

[27] W. X. Li, V. Mahadevan, N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, 2014. DOI: 10.1109/TPAMI.2013.111.

[28] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, R. Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018. DOI: 10.1016/j.cviu.2018.02.006.

[29] G. Kim, S. Lee, S. Kim. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014. DOI: 10.1016/j.eswa.2013.08.066.

[30] R. C. Aygun, A. G. Yavuz. Network anomaly detection with stochastically improved autoencoder based models. In *Proceedings of the 4th IEEE International Conference on Cyber Security and Cloud Computing*, IEEE, New York, USA, pp.193–198, 2017. DOI: 10.1109/CSCloud.2017.39.

[31] U. Fiore, F. Palmieri, A. Castiglione, A. De Santis. Network anomaly detection with the restricted Boltzmann machine. *Neurocomputing*, vol. 122, pp. 13–23, 2013. DOI: 10.1016/j.neucom.2012.11.050.

[32] W. Li, Q. Du. Collaborative representation for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1463–1474, 2015. DOI: 10.1109/TGRS.2014.2343955.

[33] P. Papadimitriou, A. Dasdan, H. Garcia-Molina. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 19–30, 2010. DOI: 10.1007/s13174-010-0003-x.

[34] C. W. Ten, J. B. Hong, C. C. Liu. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 865–873, 2011. DOI: 10.1109/TSG.2011.2159406.

[35] S. Ahmad, A. Lavin, S. Purdy, Z. Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, vol. 262, pp. 134–147, 2017. DOI: 10.1016/j.neucom.2017.04.070.

[36] T. Schlegl, P. Seebock, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Proceedings of the 25th International Conference on Information Processing in Medical Imaging*, Springer, Boone, USA, pp.146–157, 2017. DOI: 10.1007/978-3-319-59050-9_12.

[37] M. Du, F. F. Li, G. N. Zheng, V. Srikumar. DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, ACM, Dallas, USA, pp.1285–1298, 2017. DOI: 10.1145/3133956.3134015.

[38] H. M. Lu, Y. J. Li, S. L. Mu, D. Wang, H. Kim, S. Serikawa. Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2315–2322, 2018. DOI: 10.1109/JIOT.2017.2737479.

[39] P. V. Bindu, P. S. Thilagam. Mining social networks for anomalies: Methods and challenges. *Journal of Network and Computer Applications*, vol. 68, pp. 213–229, 2016. DOI: 10.1016/j.jnca.2016.02.021.

[40] W. Z. Yan, L. J. Yu. On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. https://arxiv.org/abs/1908.09238, 2019.

[41] R. M. Alguliyev, R. M. Aliguliyev, Y. N. Imamverdiyev, L. V. Sukhostat. An anomaly detection based on optimization. *International Journal of Intelligent Systems and Applications*, vol. 9, no. 12, pp. 87–96, 2017. DOI: 10.5815/ijisa.2017.12.08.

[42] M. H. Hassoun. Fundamentals of Artificial Neural Networks, Cambridge, USA: MIT Press, 1995.

[43] M. D. Tissera, M. D. McDonnell. Deep extreme learning machines: Supervised autoencoding architecture for classification. *Neurocomputing*, vol. 174, pp. 42–49, 2016. DOI: 10.1016/j.neucom.2015.03.110.

[44] R. Chalapathy, A. K. Menon, S. Chawla. Anomaly detection using one-class neural networks. https://arxiv.org/abs/1802.06360, 2018.

[45] P. Oza, V. M. Patel. Active authentication using an autoencoder regularized CNN-based one-class classifier. In *Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition*, IEEE, Lille, France, pp.1–8, 2019. DOI: 10.1109/FG.2019.8756525.

[46] S. M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, vol. 58, pp. 121–134, 2016. DOI: 10.1016/j.patcog.2016.03.028.

[47] J. An, S. Cho. Variational autoencoder based anomaly detection using reconstruction probability, Technical Report, SNU Data Mining Center, Korea, 2015.

[48] W. Li, G. D. Wu, Q. Du. Transferred deep learning for anomaly detection in hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 597–601, 2017. DOI: 10.1109/LGRS.2017.2657818.

[49] B. R. Kiran, D. M. Thomas, R. Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, vol. 4, no. 2, Article number 36, 2018. DOI: 10.3390/jimaging4020036.

[50] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, M. Ghogho. Deep learning approach for network intrusion detection in software defined networking. In *Proceedings of International Conference on Wireless Networks and Mobile Communications*, IEEE, Fez, Morocco, pp.258–263, 2016. DOI: 10.1109/WINCOM.2016.7777224.

[51] V. L. Cao, M. Nicolau, J. McDermott. A hybrid autoencoder and density estimation model for anomaly detection. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*, Springer, Edinburgh, UK, pp.717–726, 2016. DOI: 10.1007/978-3-319-45823-6_67.

[52] H. L. Yu, D. Sun, X. Y. Xi, X. B. Yang, S. Zheng, Q. Wang. Fuzzy one-class extreme auto-encoder. *Neural Processing Letters*, vol. 50, no. 1, pp. 701–727, 2019. DOI: 10.1007/s11063-018-9952-z.

[53] D. Zimmerer, S. A. A. Kohl, J. Petersen, F. Isensee, K. H. Maier-Hein. Context-encoding variational autoencoder for unsupervised anomaly detection. [Online], Available: https://arxiv.org/abs/1812.05941, 2018.

[54] M. Jeragh, M. AlSulaimi. Combining auto encoders and one class support vectors machine for fraudulant credit card transactions detection. In *Proceedings of the 2nd World Conference on Smart Trends in Systems, Security and Sustainability*, IEEE, London, UK, pp.178–184, 2018. DOI: 10.1109/WorldS4.2018.8611624.

[55] Y. S. Chong, Y. H. Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *Proceedings of the 14th International Symposium on Neural Networks*, Springer, Sapporo, Japan, pp.189–196, 2017. DOI: 10.1007/978-3-319-59081-3_23.

[56] M. Amer, M. Goldstein, S. Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ACM, Chicago, USA, pp.8–15, 2013. DOI: 10.1145/2500853.2500857.

[57] Y. C. Xiao, H. G. Wang, L. Zhang, W. L. Xu. Two methods of selecting Gaussian kernel parameters for one-class SVM and their application to fault detection. *Knowledge-Based Systems*, vol. 59, pp. 75–84, 2014. DOI: 10.1016/j.knosys.2014.01.020.

[58] I. Irigoien, B. Sierra, C. Arenas. Towards application of one-class classification methods to medical data. *The Scientific World Journal*, vol. 2014, Article number 730712, 2014. DOI: 10.1155/2014/730712.

[59] H. Yu. SVMC: Single-class classification with support vector machines. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, ACM, Acapulco, Mexico, pp.567–572, 2003.

[60] M. Hejazi, Y. P. Singh. One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence*, vol. 27, no. 5, pp. 351–366, 2013. DOI: 10.1080/08839514.2013.785791.

[61] W. Khreich, B. Khosravifar, A. Hamou-Lhadj, C. Talhi. An anomaly detection system based on variable N-gram features and one-class SVM. *Information and Software Technology*, vol. 91, pp. 186–197, 2017. DOI: 10.1016/j.infsof.2017.07.009.

[62] C. Gautam, R. Balaji, K. Sudharsan, A. Tiwari, K. Ahuja. Localized multiple kernel learning for anomaly detection: One-class classification. *Knowledge-based Systems*, vol. 165, pp. 241–252, 2019. DOI: 10.1016/j.knosys.2018.11.030.

[63] B. Krawczyk, M. Wozniak, B. Cyganek. Clustering-based ensembles for one-class classification. *Information Sciences*, vol. 264, pp. 182–195, 2014. DOI: 10.1016/j.ins.2013.12.019.

[64] D. M. J. Tax, K. R. Muller. Feature extraction for one-class classification. In *Proceedings of Joint International Conference ICANN/ICONIP*, Istanbul, Turbey, pp.342–349, 2003. DOI: 10.1007/3-540-44989-2_41.

[65] Y. Goldberg, O. Levy. word2vec explained: Deriving Mikolov et al.′s negative-sampling word-embedding method. [Online], Available: https://arxiv.org/abs/1402.3722, 2014.

[66] L. Van Der Maaten, G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[67] E. Mayoraz, E. Alpaydin. Support vector machines for multi-class classification. In *Proceedings of the International Work-conference on Artificial Neural Networks*, Springer, Alicante, Spain, pp.833–842, 1999. DOI: 10.1007/BFb0100551.

[68] C. Zhou, R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Halifax, Canada, pp.665–674, 2017. DOI: 10.1145/3097983.3098052.

[69] L. Manevitz, M. Yousef. One-class document classification via neural networks. *Neurocomputing*, vol. 70, no. 7–9, pp. 1466–1481, 2007. DOI: 10.1016/j.neucom.2006.05.013.

Abhijit Guha received the B. Sc. degree (Chemistry Honors) from Calcutta University, India in 2006, and MCA (master of computer applications) degree in computer applications degree from Academy of Technology under West Bengal University of Technology, India 2009. He is a Ph. D. degree candidate in Department of Data Science, CHRIST (Deemed to be University), India. Presently, he is working as a research and development scientist in First American India Private Limited. His research areas include document image processing, data mining, statistical modeling, machine learning modelling in title insurance domain. He has delivered multiple business solutions using the AI technologies and received consecutive three "Innovation of the year" awards from 2015 to 2017 by First American India for his contribution towards his research.

His research interests include artificial intelligence, natural language processing, text mining statistical learning and machine learning.

E-mail: abhijitguha.research@gmail.com (Corresponding author)

ORCID iD: 0000-0002-3280-5730

Debabrata Samanta received the B. Sc. degree (Physics Honors) from Calcutta University, India in 2007, and MCA degree from Academy of Technology under West Bengal University of Technology, India in 2010, and the Ph. D. degree in computer science and engineering from National Institute of Technology, India in 2014. In 2015, he was a faculty member at Dayananda Sagar University, India and in 2019 he was at CHRIST (Deemed to be University), India. Currently, he is an assistant professor in Department of Computer Science at CHRIST (Deemed to be University), India. He is a professional IEEE member, an associate life member of Computer Society of India (CSI) and a life member of Indian Society for Technical Education (ISTE). He has authored and coauthored over 127 papers in SCI/Scopus/Springer/Elsevier journals and IEEE/Springer/Elsevier conference proceedings in areas of artificial intelligence, natural language processing and image processing. He has received "Scholastic Award" at the 2nd *International conference on Computer Science and IT Application*, CSIT-2011, India. He has published 9 books, available for sale on Amazon and Flipkart. He has edited 1 book available on Google Book server. He has authored and coauthored of 2 Elsevier and 5 Springer Book Chapter. He is a convener, keynote speaker, technical programme committee (TPC) member in various conferences/workshops, etc. He was an invited speaker at several Institutions.

His research interests include artificial intelligence, natural language processing and image processing.

E-mail: debabrata.samanta369@gmail.com

ORDID iD: 0000-0003-4118-2480

International Journal of Automation & Computing

Citation: A. Guha, D. Samanta. Hybrid approach to document anomaly detection: an application to facilitate rpa in title insurance. *International Journal of Automation and Computing*, vol.18, no.1, pp.55–72, 2021. https://doi.org/10.1007/s11633–020–1247–y