Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 6681886, 15 pages
https://doi.org/10.1155/2023/6681886

Research Article

Forecasting the Friction Coefficient of Rubbing Zirconia Ceramics by Titanium Alloy

Ahmad Salah,[1,2] **Ahmed Fathalla**,[3] **Esraa Eldesouky**,[4,5] **Wei Li**,[6] **and Ahmed Mohamed Mahmoud Ibrahim**[6,7,8]

[1]*College of Computing and Information Sciences, University of Technology and Applied Sciences, Ibri, Oman*
[2]*Faculty of Computers and Informatics, Zagazig University, Zagazig, Sharkia, Egypt*
[3]*Department of Mathematics, Faculty of Science, Suez Canal University, Ismailia, Egypt*
[4]*Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*
[5]*Department of Computer Science, Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt*
[6]*College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China*
[7]*Department of Mechanical and Aerospace Engineering, College of Engineering, United Arab Emirates University, Al Ain 15551, UAE*
[8]*Production Engineering and Mechanical Design Department, Faculty of Engineering, Minia University, Minya 61519, Egypt*

Correspondence should be addressed to Ahmed Fathalla; fathalla_sci@science.suez.edu.eg and Wei Li; liwei@hnu.edu.cn

Received 5 June 2023; Revised 25 October 2023; Accepted 11 November 2023; Published 12 December 2023

Academic Editor: Alexander Hošovský

The thermal issues generated from friction are the key obstacle in the high-performance machining of titanium alloys. The friction between the workpiece being cut and the cutting tool is the dominant parameter that affects the heat generation during the machining processes, i.e., the temperature inside the cutting zone and the consumed cutting energy. Besides, the complexity is associated with the nature of the friction phenomenon. However, there are limited efforts to forecast the friction coefficient during the machining operations. In this work, the friction coefficients between the titanium alloy against zirconia ceramics lubricated by minimum quantity lubrication were recorded and measured using a universal mechanical tester pin-on-disc tribometer. Then, we proposed two models for forecasting the friction coefficient which are trained and tested on the recorded data. The two predictive models are based on autoregressive integrated moving average and gated recurrent unit deep neural network methods. The proposed models are evaluated through a set of exhaustive experiments. These experiments demonstrated that the proposed models can efficiently be used to reduce power consumption dedicated to monitoring the friction coefficients. Besides, they can reduce or avoid surface thermal damage by predicting the high level of friction coefficients in advance, which can be used as an alert to enable or readjust the lubrication parameters (fluid pressure, fluid flow rate, etc.) to maintain lower ranges of friction coefficients and power consumption.

1. Introduction

One of the main reasons for energy wasting in milling machines is friction [1]. Friction is the key reason for the heat generated during the metal cutting process [2, 3]. However, friction is inherent to any rubbing surface. It is usually uncontrolled and undesirable due to the corresponding collateral damages regarding the tool deteriorations, surface thermal damages, unwelcome energy consumption, etc. It is scientifically proven that about 20% of the lost energy in the world is consumed through friction [4]. Thus, building a predictive model to forecast the behavior of friction experiments is crucial in saving the time and cost of such experiments (e.g., universal mechanical tester (UMT) pin-on-disc universal tribometer). The UMT pin-on-disc universal tribometer is one of the wide range of machines that are used to perform frictional tests.

Nonetheless, friction as a phenomenon is greatly affected by many parameters such as the applied load, sliding velocity, the rubbing materials, the temperatures, and the humidity. [5, 6]. Therefore, predicting the friction coefficient is a complicated mission. Thus, an efficient model of predicting the friction coefficients based on machine learning or deep learning can represent the key in estimating the cutting tool operating life, estimating the critical contact temperature (generated from the friction) to avoid thermal damages, maintaining the sustainability of the energy by keeping the friction coefficient in specific ranges, etc.

The efforts to predict the parameters related to the friction experiments have been conducted in a few pieces of research [7–9]. All of these efforts include a numerical method to simulate friction experiments. To the best of the authors' knowledge, there is no research conducted to predict the friction coefficient and energy consumption during the machining operations or no research utilized machine learning techniques for the sake of friction coefficient prediction. An efficient friction coefficient prediction model can reduce the machining costs by decreasing the number of defective products due to thermal damages and the tooling by extending the operation life. Furthermore, energy consumption is inherent to the friction coefficient. Thus, maintaining the friction coefficients within specific limits can decrease energy consumption significantly, which is reflected positively in the economy, environment, and energy savings. These issues motivate the current work. All of these points motivated this work to predict the friction coefficients.

There are several applications for forecasting the friction coefficient in a cutting machine [6, 10]. For instance, a lubrication system integrated into the machining cycle could be implemented. A closed-loop control system could be used to control the friction coefficient within the safe limit. The corresponding forecasting model is responsible for evaluating the friction coefficient and forecast whenever the friction coefficient exceeds the safe limit. The control system can readjust the lubrication parameters (lubricant pressure, flow rate, etc.) to lessen the friction before the failure of the tool and the workpiece.

In another example, forecasting the friction coefficient can help in reducing the power consumption of the experiment. For instance, if the cutting machine should work for 20 minutes, then the force sensor should work for 20 minutes as well to monitor the friction coefficient. Using the forecast model, the force sensor and the forecast model can work interchangeably (i.e., sensor force works for only ten minutes); thus, the time the forecast model is working the force sensor can be turned off. In addition, turning off the force sensor for a while and then reusing this sensor improves the sensor's accuracy.

In this context, we proposed framing the task of forecasting friction coefficient in the UMT pin-on-disc universal tribometer experiments as a time series forecasting task. The friction coefficient changes over time in frictional experiments. Thus, we proposed designing models to capture the behavior of changes in the friction coefficient values over time and then forecast future values. In the first step, we generated a dataset of friction coefficient values. These values are obtained by attaching force sensors to the UMT pin-on-disc tribometer and then an experiment was conducted to collect the friction coefficient values at different time points (e.g., every millisecond).

In this vein, we proposed designing a predictive model of the friction coefficient of milling machines. We started by recording the friction coefficient between the titanium alloy (i.e., Ti-6Al-4V) against zirconia ceramics (i.e., ZrO_2). Then, we proposed utilizing the autoregressive integrated moving average (ARIMA) statistical model for forecasting purposes. The main drawback of the statistical models is that there is no mechanism for updating these models. Therefore, if there are new patterns in the data, then the statistical model should be rebuilt. On the other hand, deep learning- (DL-) based forecasting models support updating the model's weights as the data pattern changes. Thus, we proposed harnessing the DL methods due to its capability of updating and identifying nonlinear and complex patterns in different domains (e.g., weather forecasting [11], energy markets [12], and e-commerce products [13, 14]), where historical observations of a variable are analyzed to introduce a model to describe the underlying relationship. The proposed DL-based model is built using the gated recurrent unit deep neural network (GRU-DNN) architecture, a variation of the well-known recurrent neural network (RNN) architecture, which has shown impressive performance in the time series forecasting field [15, 16].

The contributions of this work can be summarized as follows:

(1) To our knowledge, we proposed the first DL-based model (i.e., GRU-DNN model) and the first statistical-based model (i.e., ARIMA model) for forecasting the friction coefficient in cutting machines. The proposed GRU-DNN architecture and ARIMA model's parameters were designed to fit friction coefficient data patterns. This model is utilized in scenarios where the forecast model needs to be updated.

(2) This work provides a publicly available friction coefficient dataset that can be used to improve the task of predicting friction coefficient.

(3) The proposed forecasting models are evaluated thoroughly using four different scenarios on four different evaluation metrics. The obtained solutions outlined the efficiency of the proposed solution to forecast accurate friction coefficient values.

The rest of the paper is organized as follows. Section 2 discusses the existing machine learning-based methods utilized for handling the predictive tasks related to metal cutting tools. In Section 3, the background is discussed. Section 4 exposes the proposed methodology. The evaluation of the proposed prediction model is presented in Section 5. Discussion is shown in Section 6. Finally, the paper is concluded in Section 7.

2. Related Work

This section discusses the research efforts undertaken to explore phenomena associated with cutting tools. The employment of machine learning methods in these attempts demonstrates variability, with some studies using such algorithms and others do not [17–19]. These phenomena have a substantial impact on various difficulties, including the reduction of dimensional accuracy in the cut surface, tool breakage, and machine downtime. Song et al. [17] proposed a predictive model for estimating cutting forces in carbon fiber reinforced polymer (CFRP) materials based on nonlinear regression analysis. Saha et al. [18] presented an energy-based model for the prediction of cutting forces in machine operations. Authors aimed to examine the factors that contribute to the onset of adhesion in the context of progressive tool wear. Geng et al. [20] introduced an enhanced predictive model for estimating the torque and thrust force for the GJV450 under conditions of elevated temperature, big deformation, and significant strain rate. The proposed model relies on a calculus approach to depict the variation of partial working angles during tool machining.

Recently, machine learning algorithms have been widely recognized for their capability to address nonlinear and complex interactions by means of data training. Several studies have used machine learning algorithms in tribological research to analyze and assess various phenomena, such as cutting force prediction or cutting tool lifetime, to improve machine efficiency [21]. The thermal effects on machine operations have been predicted using machine learning models [22, 23].

Saravanan et al. [22] presented a nonmechanical engineering model that included neural approaches, namely, logistic regression (LR), k-nearest neighbors (KNN), and random forest (RF). The primary objective is to forecast the thermal efficiency of a c-shaped finned solar air heater (SAH). Meanwhile, Zhang et al. [8] proposed an analytical model to investigate coated cutting tools' rake face temperature distribution in the machining of H13 hardened steel. Furthermore, Singh et al. [23] examined the machine learning ability to forecast the maximum temperature in an elastohydrodynamic lubricated (EHL) line contact. The operation of the EHL system under excessive loads or rates can elevate the temperatures, hence increasing the likelihood of unexpected system failure [24]. The integration of a neural network model with a machine learning model results in a significantly high level of accuracy, approximately 0.998. This integration also enhances the model's capability to effectively record the nuances of the EHL system.

The authors in [25] used supervised machine learning regression-based techniques to make predictions on the ultimate strength friction (USF) stir of stir welded magnesium joints. The XGboost algorithm has exhibited superior accuracy, as evidenced by its coefficient of determination value of 0.816. The XGboost algorithm outperformed other machine learning models such as DT, RF, and AdaBoost. However, additional investigation is required in order to comprehensively comprehend the influence of other output parameters on stir welded joints, not limited to the USF output parameter. Furthermore, the authors of [26] utilized various machine learning techniques to make predictions regarding the collective characteristics, including tool failure diagnostics, and real-time control of friction stir welding (FSW).

The authors in [27] conducted a series of experiments on an annealed Ti-6Al-4V alloy. The aim was to assess the efficacy of their suggested machine learning approach for predicting cutting force. The authors used a support vector machine (SVM) classifier with a polynomial kernel to ascertain the correlation between the properties of the cutting force signal and the wear of the tool. The classifier's accuracy and F1-score rates were 91.43% and 86.94%, respectively, as reported in the study.

Diaz-Rozo et al. [28] devised a diagnostic tool to evaluate the performance of machined spindles using three clustering algorithms. The present study focused on the analysis of machine spindle behavior patterns through the utilization of clustering algorithms in an unsupervised manner. By examining the collected instance data, significant information on spindle data can be extracted and the spindle data is partitioned into distinct groups based on their inherent characteristics. Krishnakumar el al. [29] proposed a tool monitoring system using a classification model to categorize and monitor tool conditions in a high-speed precision milling center. Statistical features such as count, amplitude, and mean are captured using an acoustic emission sensor signal. Hence, the dominant features with maximum entropy are selected for classification using DT and SVM models.

Lawrence et al. [30] examined the impact of tool vibration and cutting parameters on buffered impact dumpers- (BID-) assisted boring process using artificial neural networks (ANNs). The ANN model demonstrated a high level of accuracy in predicting various aspects of cutting performance, including surface roughness, tool vibration, and cutting force. The experimental results indicated a significant reduction in the cutting force, surface finish, and tool vibration by 85%, 95%, and 93%, respectively.

The prediction of cutting tool lifetime and torque values relies on the use of torque data as a dependable indicator [31]. Oberlé et al. [31] developed a regressor model to forecast tool wear along with utilizing recorded torque data from the machining center and then measuring tool wear directly. The performance of the regressor model was evaluated by using the random forest (RF) technique, resulting in an R2 score of 74%. In [32], the tool wear throughout the machining process was predicted using a conventional neural network (CNN) approach.

2.1. Gap Analysis.

The discussion of the literature on friction coefficient prediction outlines the current efforts to use different machine learning approaches to several related problems, such as predicting tool wear and tool lifetime. Despite these efforts, we did not find any literary evidence proposing the prediction of the friction coefficient during metal cutting procedures using any learning-based model. Thus, utilizing deep learning and statistical models is required to investigate their accuracy levels.

3. Background

3.1. Parameters Affecting the Friction.

In the following text, the parameters that noticeably affect the friction are discussed. Since the nanofluid is delivered to the contact area in the form of droplets. Thus, the contact angle and the surface tension of the droplets control the efficiency of the lubrication. For instance, at GNPs content of 0.03, the contact angle jumps to 34.475° [1]. Higher GNPs contents of 0.1, 0.2, 0.3, and 0.4 wt.%, which recorded 36.89°, 40.73°, 43.47°, and 48.51°, displayed the same pattern. This trend is explained by nanoparticles' propensity to congregate in nanofluid droplets in larger concentrations. As a consequence, the surface free energy increases [33]. Additionally, as concentrations rise, the surface-to-volume ratio rises as well. As a result, the electrostatic repulsive force is outweighed by the attractive Van der Waals force, which raises surface energy and contact angle [34, 35]. As a result, with larger GNPs concentrations, the nanofluids' wettability region shrinks. The GNPs reduce the impact of the higher contact angle and maintain a strong antifriction layer between the workpiece and the grinding wheel. The normal and tangential forces are significantly decreased as a result of the antifriction layer that the PG0.03 droplets create. Additionally, the same general behavior is observed for all nanofluids derived from palm oil with GNP levels ranging from 0.03 to 0.4 weight percent. However, the increased contact angles and relatively higher viscosity of the palm oil-based GNPs nanofluids prevent them from spreading easily across the surface of the workpiece. Because of this, there is direct contact between the grinding wheel and the workpiece in some areas. The machined surface's surface roughness rises as a result of this close contact. The applied load plays an unspecified role in the friction. Therefore, the role of applied load is usually selected based on the application and frictional test standards [36]. Interestingly, the lubrication medium is a dominant parameter in the friction operation, since the used medium is palm oil which is a member of class of oils known as triglyceride and fatty acid composing oils. The strong polarity of palm oil makes it a great lubricant (−COOH in fatty acids and −COOR in triglycerides) [37].

In this section, we briefly review the two forecasting models used in the proposed forecasting problem. Two classes of forecasting methods, namely, a statistical approach (i.e., ARIMA model) and a DL-based approach (i.e., GRU-DNN model), are proposed for forecasting the friction coefficient of milling machines.

3.2. ARIMA.

ARIMA model is a popular and widely used linear model in time series forecasting [38], thanks to its statistical properties and the Box–Jenkins methodology [39] in its building process. The ARIMA model is composed of three different types of time series, namely, pure autoregressive (AR), pure moving average (MA), and the integration of AR and MA (ARMA) series. Thus, for a p^{th} order autoregressive model, (AR_p) is given by equation (1). Similarly, for a q^{th} order moving average model, (MA_q) is given by equation (2).

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + \epsilon_t, \tag{1}$$

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}, \tag{2}$$

where Y_t is the dependent (response) variable at time point t, $Y_{t-1}, Y_{t-2} \ldots Y_{t-p}$ are the response variable at time point lags $t-1, t-2, \ldots, t-p$, respectively, $\beta_0, \beta_1 \ldots \beta_p$ are coefficients to be evaluated, $\epsilon_t, \epsilon_{t-1} \ldots \epsilon_{t-p}$ are error terms at time $t, t-1, t-2, \ldots, t-p$, respectively, and μ is the constant mean of the process.

In the ARIMA model, a variable future value is considered to be a linear function of several past observations and random errors. Specifically, the whole process that defines the time series has the form that is represented as follows:

$$\begin{aligned} Y_t = {} & \gamma_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \ldots + \beta_p X_{t-p} \\ & + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \ldots - \theta_q \epsilon_{t-q}, \end{aligned} \tag{3}$$

where γ_0 is a coefficient to be estimated. Random errors (ϵ_i) are supposed to be independent and distributed identically such that

$$\begin{aligned} E(\epsilon_i) &= 0, \\ \text{var}[\epsilon_i] &= \sigma^2. \end{aligned} \tag{4}$$

Typically, we assumed

$$\epsilon_i \sim N(0, \sigma^2). \tag{5}$$

3.3. Gated Recurrent Unit Neural Networks.

Recurrent neural network (RNN) is one of the deep learning methods that have been widely used in a range of applications successfully. In specific, it has been applied to time series forecasting. The RNN has been utilized to successfully address several problems [40–42]. It is a robust model that can learn a wide range of complex associations through vast amount of data. However, the depth of RNN results in two well-known problems, exploding and vanishing gradient problems. Therefore, two variations of the recurrent model were introduced, i.e., GRU [43] and LSTM [44], to address the aforementioned problems that arose with the vanilla RNN.

LSTM and GRU architectures are characterized by similarity in design which include a gating mechanism for regulating the information flow through the unit. Nevertheless, due to the complex structure of the LSTM, its training and converging time is too long. GRU-DNN has simpler architecture compared to LSTM; thus, a GRU-DNN model is faster to train, relative to an LSTM model [45].

The GRU model was introduced to allow recurrent units to capture patterns and dependencies of different time scales. Compared to the LSTM cell, GRU has no separate memory gate, which makes it more efficient and fast in data training. Figure 1 depicts a standard cell architecture for a GRU model. A typical GRU is composed of a group of cells in which each cell includes two gates (i.e., update $z_{(t)}$ and reset $r_{(t)}$) and a hidden state vector denoted as $h_{(t)}$ over the current time point t. A neural network of one layer is used to express each gate. A GRU cell architecture can be better illustrated in the light of the following equations: equations (6)–(9). The cell is fed by two values, namely, the preceding hidden state cell and the current input sequence vector which is denoted as $h^{(t-1)}$ and $x^{(t)}$, respectively. The cell output $h^{(t)}$ stands for a hidden state.

$$z^{(t)} = \sigma\left(W_z x^{(t)} + U_z h^{(t-1)} + b_z\right), \tag{6}$$

$$r^{(t)} = \sigma\left(W_r x^{(t)} + U_r h^{(t-1)} + b_r\right), \tag{7}$$

$$\widetilde{h}^{(t)} = \tan h\left(W_h x^{(t)} + U_h\left(h^{(t-1)} \odot r^{(t)}\right) + b_h\right), \tag{8}$$

$$h^{(t)} = z^{(t)} \odot h^{(t-1)} + \left(1 - z^{(t)}\right) \odot \widetilde{h}^{(t)}, \tag{9}$$

where $\odot$ represents an element-wise multiplication, $\sigma(.)$ and $\tan h(.)$ are the sigmoid and hyperbolic tangent activation functions of the NN, respectively, $\widetilde{h}^{(t)}$ represents the candidate hidden state, W_z, W_r, and W_h are the cell model's weight matrices for the feed-forward neural networks, while U_z, U_r, and U_h are the cell model's weight matrices for the recurrent neural network, and b_z, b_r, and b_h are the model bias.

A GRU cell's output $h^{(t)}$, given by equation (9), forms a linear interpolation between a current candidate state $\widetilde{h}^{(t)}$ and a previous hidden state $h^{(t-1)}$. This linear interpolation mechanism is mainly used for learning long-term dependencies. More precisely, as $z^{(t)}$ approaches 1, the preceding hidden state remains unaltered and is kept unchanged for a few successive time steps. On the contrary, as $z^{(t)}$ approaches 0, the output cell will be equal to the value of the candidate state $\widetilde{h}^{(t)}$. In this case, the candidate state's value $\widetilde{h}^{(t)}$ depends highly on the current input and the previous hidden state. Similarly, the candidate state $\widetilde{h}^{(t)}$ depends also on the reset gate $r^{(t)}$, which forces the cell to exclude or keep the last hidden states.

4. Methodology

4.1. The Proposed ARIMA Model. The stationarity property of a data series is a mandatory condition for building an efficient ARIMA model which is able to forecast. A time series is stationary when its statistical characteristics (i.e., mean and variance) are constant over time. Furthermore, the ARIMA model has different parameters that are required to be estimated or tuned for building an efficient forecasting model.

Nonseasonal ARIMA parameters are p, d, and q, where p denotes the order of the autoregressive part, d is the required number of differencing needed to convert the data series into stationary series, and q is the moving average order. Additionally, if the data series has a seasonality trend, then more seasonal parameters should be considered by the model. Therefore, to obtain the ARIMA parameters (nonseasonal parameters e.g., p, d, and q, and seasonal parameters e.g., P, D, Q, and S), we perform a grid search (using auto_arima method (https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html)) over a search space, as presented in Table 1. In other words, several model parameter combinations are evaluated; meanwhile, the model parameter combination that achieves the best fitting score is reported.

4.2. The Proposed Stacked GRU-DNN Model. The proposed stacking GRU-DNN is depicted in Figure 2. The model architecture consists of an input layer, a GRU layer, a fully connected layer (FC), and an output layer. The input layer accepts the model input whereas the output layer involves one neuron to produce the predicted value. The primary intention for using such a model structure is to utilize a recurrent layer that has the ability to learn and model time series patterns in the dataset. However, the intermediate fully connected layers are beneficial for recombining the extracted representation acquired from preceding layers and gaining supplementary representations for higher levels of abstraction.

Neural network models are prone to overfitting or underfitting problems, which are caused by the excessive/less training epochs of the neural network model [46]. Therefore, one possibility for resolving the overfitting or underfitting problems in DL-model is to apply an early halting strategy. The training is devoted in using this strategy when generalization performance commences degrading over a successive number of epochs. As a consequence, to follow up on the generalization performance, the training data is split into training and validation groups.

Another method to tackle the overfitting problem is to use the dropout method [47]. Dropout is a regularization method that permits training neural networks with different architectures in parallel, where a certain ratio of layer neurons are randomly ignored or dropped out. Dropout is represented in the fully connected layers by the black neurons as shown in Figure 2.

Adam optimizer [48], which is an adaptive optimization algorithm, is used with its default learning and decay rate settings. Adam optimizer has demonstrated its efficiency in solving practical DL problems, and its results outperform the other stochastic optimization methods. The proposed DL model uses the mean square error (MSE) loss function, given by equation (10), that is, given a training data $\{(X_i, Y_i)\}_{i=1}^{N}$ of N observations, the proposed GRU-DNN model is trained with the objective of minimizing the following loss function:

$$\min_{w} \frac{1}{N} \sum_{j=1}^{N} \left\{ \left\| Y_j - F\left(X_j, w\right) \right\|_2 \right\}, \tag{10}$$

where w is the network coefficient, F: $\mathbb{R}^k \longrightarrow \mathbb{R}^1$ denotes the neural network flow, and k is the input vector size (i.e., number of lag features).

4.2.1. The Proposed GRU-DNN Model's Hyperparameter Optimization. Machine learning algorithms involve the optimization of model hyperparameters. Hyperparameters refer to the model parameters (coefficient) that are used to control the training task. Thus, such parameters (e.g., number of layers/neurons of a network/layer, learning rate, and lag order of ARIMA model) should be fine-tuned prior to the forecasting process. Hyperparameters tuning (or optimization) refers to the process of obtaining the best values for a set of hyperparameters that results in good fitting/generalization of the model. In our proposed work, obtaining the best model hyperparameters is achieved using

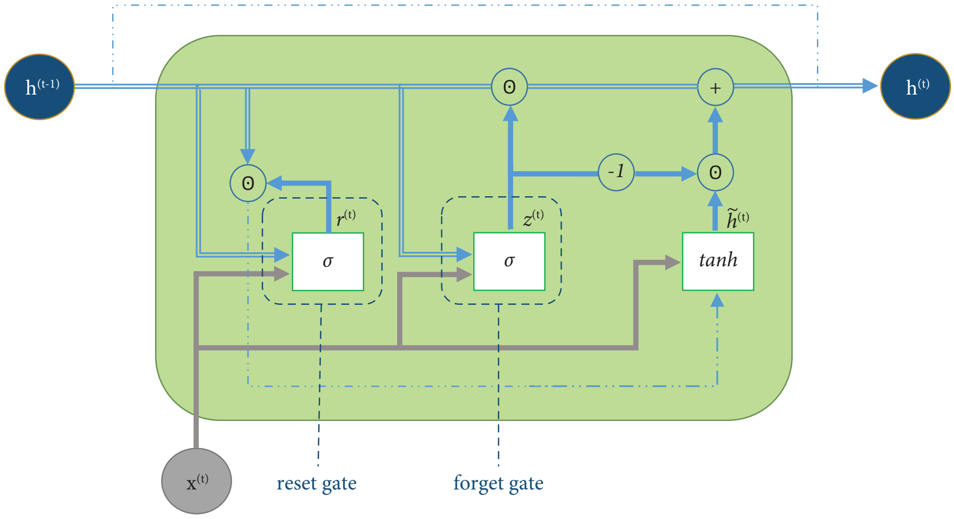

Figure 1: GRU cell architecture.

Table 1: The hyperparameter search space for the proposed ARIMA model.

Hyperparameters	Value
p	[0, 1, 2, 3, 4, 5]
d	[0, 1, 2, 3, 4, 5]
q	[0, 1, 2, 3, 4, 5]
P	[1, 2]
D	[0, 1]
Q	[1, 2]

an asynchronous distributed hyperparameter optimization method [49]. Moreover, regarding parameter searching and optimization, we utilized the Tree-structured Prazen Estimator (TPE) methodology [50] from the Hyperopt package (https://hyperopt.github.io/hyperopt/). Table 2 shows the GRU-DNN model hyperparameters and the search spaces applied to obtain the optimal hyperparameter values of the model.

Figure 3 depicts the required steps to build the proposed GRU and ARIMA models, starting from collecting the data to training the proposed models up to the evaluation phase. It is worth pointing out that for building the ARIMA model, the validation data set is not used; therefore, the validation data are appended to the training set.

5. Experimental Results

5.1. Experimental Setup. Framework implementation is developed in the Python programming language. Moreover, the dataset is loaded using the Pandas [51] DataFrame. The proposed statistical model (e.g., ARIMA/SARIMA) utilized the *statsmodel* package [52], which provides implementations for estimating most of the statistical models. Furthermore, the proposed stacked GRU-DNN model is implemented using Keras (https://keras.io) Tensorflow [53] libraries. Moreover, other used libraries include Hyperopt [50], Scikit-Learn [54], Numpy [55], and Matplotlib [56].

The source code of the proposed work is freely accessible online on the author's GitHub website (https://github.com/Ahmed-Fathalla/Friction_coef-forecasting) in order to guarantee the reproducibility of the experimental models, parameter configurations, and reported results.

5.2. Dataset. The sliding tests were done via the universal pin-on-disc tribometer standards (https://www.astm.org/Standards/G133.htm). Besides, the sliding tests were done between Ti alloy and Zirconia ball as a counter ball. The whole sliding tests were accomplished at a sliding speed of 0.1 m/s (reciprocating speed) and 10 N as an applied load. The tests were done at lubricating conditions of dry, LB2000, water GNPs, PG-0.0, PG-0.03, PG-0.1, PG-0.2, PG-0.3, and PG-0.4. The whole tests were done for 12,000 cycles/1,200 cycles per second. Furthermore, the whole tests were done at room temperature (25°C) and humidity of 72%. The frictional coefficients were obtained automatically from the machine by dividing the tangential force over the normal force (applied force). The fluids were delivered to the cutting zone using MQL. The chemical and physical properties of the used material are shown in Tables 3 and 4, respectively.

The main target of the study is to apply the proposed technique in the field of cutting processes, especially in grinding processes. The contact between the machined surface and the cutting edges of each particle (the cutting tool is the abrasive edges) is considered a point contact. Thus, the applied load of 10 N has been chosen to simulate the same Hertizian stresses as the stresses generated by the abrasive edges [57]. On the other hand, the sliding speed of 0.1 m/s has been chosen to simulate the linear federate during the grinding operation. Furthermore, the whole friction experiment and the applied parameters have been chosen according to the standards [36].

The rubbing tests were done on a Ti-6Al-4V sample with a dimension of 25 × 12 mm and a counter 9.5 mm diameter ball of zirconia ball using UMT pin-on-disc universal tribometer with the ASTM standards [36], as shown in Figure 4.

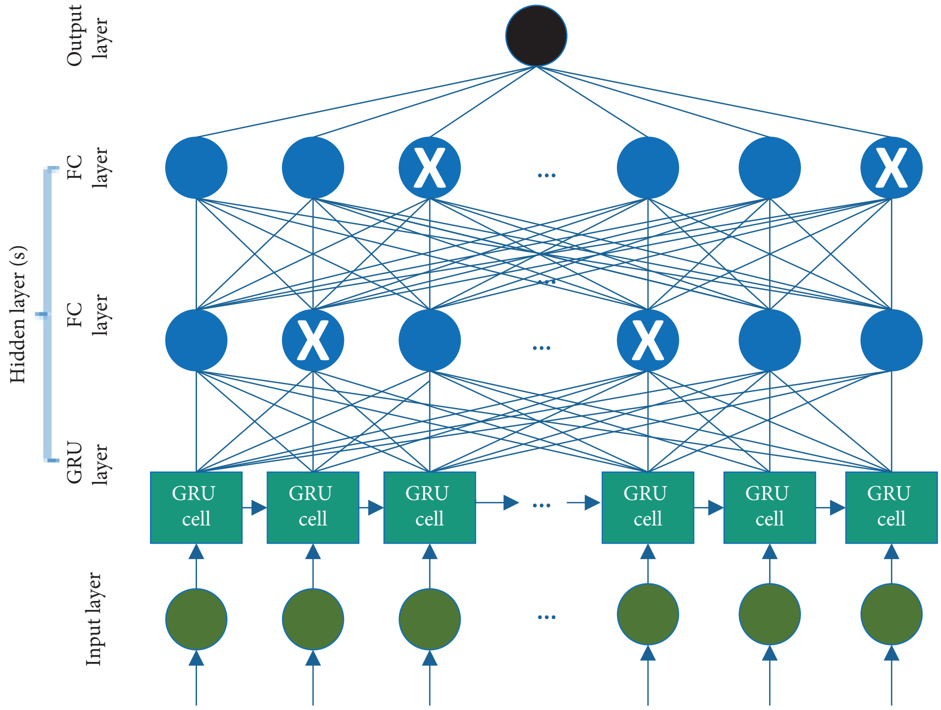

Figure 2: The proposed stacked GRU-DNN model.

Table 2: The hyperparameter search space of the proposed GRU-DNN model.

Hyperparameters	Value
No. of GRU cells	[4, 8, 16]
No. of FC layers	[1, 2]
No. of FC layers' units	[4, 8, 16]
Hidden layers activation	(ReLU, linear)
Batch size	[4, 8, 16]
Dropout rate of FC layers	[0.0, 0.1, 0.2]

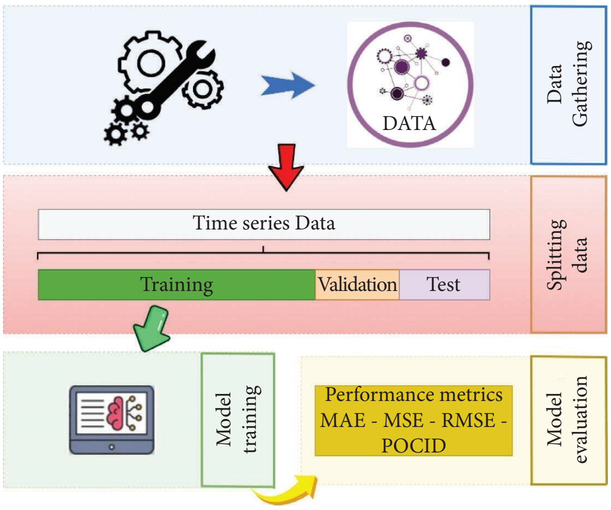

Figure 3: The block diagram of the proposed method.

The Ti-6Al-4V sample was kept fixed while the counter ball was reciprocated with a sliding speed of 0.1 m/s under an applied load of 10 N. The tribopairs were dried with hot air and cleaned with acetone to remove any existing impurities. Besides, the fluids were delivered to the contact zone as aerosols with minimum quantity lubrication technique. For the nanofluid base, we utilized palm oil and distilled water with the chemical compositions shown in Table 3.

The experiments were performed against a commercial grade 5 (as a workpiece) alloy of Ti-6Al-4V given by Dongguan Luyuan Metal Material Co., Ltd., while the cutting tool was described as a ZrO2 ball. The chemical composition of both the workpiece and the counter ball are shown in Table 4 while GNPs with technical details are shown in Table 5. The GNPs were combined with palm oil and distilled water, using various graphene levels, as listed in Table 6, to prepare the nanofluids.

The mixtures were mechanically mixed for 1 hour so that nanoadditives were homogeneously dispersed. Nonetheless, for four hours, mixtures were sonicated at 40 kHz frequency at a temperature of 35°C to prevent agglomerations and nanoadditives sedimentation and preserve a stable GNP suspension in the palm oil. However, the preparation route differentiates in the case of distilled water-based nanofluids due to the absence of the polar heads which affect the stability of the dispersion of the nanoadditives in the distilled water. Therefore, sodium deoxycholate (SDOC) with a content percentage of 0.46 mg/ml was added as a surfactant and mechanically mixed with the distilled water for 10 min. Next, ethanol with a percentage of 10 wt.% was appended to the solution and mechanically stirred for 20 min to enhance the dispersion of the nanoadditives in the distilled water and decrease the possibility of sedimentations. GNPs additives were finally added to the solution and sonicated for 24 hours at constant ambient temperature (25°C) and 40 kHz of frequency.

Table 3: Palm oil fatty acids contents and physical properties.

Physical properties			
Pour point (°C)	Flash point (°C)	Dynamic viscosity (Pa.s)	
		RT (25°C)	60°C
23.6	314	$0.07144 \pm 3.79 \times 10^{-4}$	$0.02223 \pm 7.213 \times 10^{-5}$

Chemical composition				
Saturated fatty acids			Monounsaturated fatty acids	Polyunsaturated fatty acids
Palmitic (C16 : 0)	Stearic (C18 : 0)	Myristic (C14 : 0)	Oleic (C18 : 1)	Linoleic (C18 : 2)
44.3%	4.6%	1.0%	38.7%	10.5%

Table 4: Ti alloy and zirconia ball chemical and physical properties.

Ti-6Al-4V alloy "grade 5"							
Chemical composition							
Al	V	C	Fe	O	N	H	Ti
6%	4%	0.03%	0.1%	0.15%	0.01%	0.003%	Balance
Density (gm/cm^3)							
4.57							
Hardness (HRC)							
32							
Zirconia ball (ZrO2)							
Density (gm/cm^3)							
6.02							
Hardness (HRC)							
77							

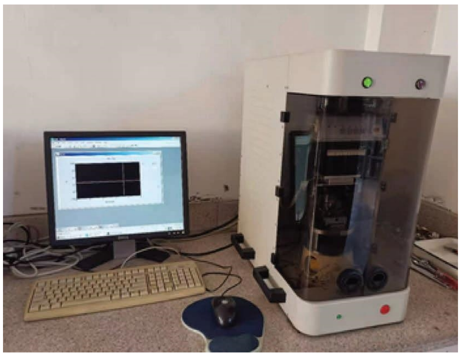

Figure 4: Universal pin-on-disc tribometer.

Table 5: Technical details of GNPs.

Diameter	5–10 mm
Thickness	3–10 nm
Surface area	31.657 m^2/g
Tap density	0.075 g/cm^3
Apparent density	0.050 g/cm^3
Purity	> 99.5%

5.2.1. Data Preparation. The dataset consists of a number of observations that are gathered over 14.76 minutes, where the observations are collected every 0.01 seconds. Thus, the dataset has $14.7645 \times 60 \times 100 = 88,587$ observations. In order to evaluate the proposed models, the dataset is divided into a train set and a test set while the test set results are reported. Furthermore, for deep learning models, data series are standardized (or normalized), resulting in data of zero mean and scaling to unit variance (given by equation (11)), which leads to better training and fast convergence of neural networks [58], before training, and the predicted values are back-transformed into original scale.

$$y_i = \frac{y_i - \mu}{\sigma}, \tag{11}$$

where μ is the data series mean and σ is its standard deviation.

5.3. Experiments. The evaluation of the proposed models is achieved by performing a set of four classes of experiments with different objectives and configurations. All of the reported results are the average of running the proposed forecast models on the obtained ten files, as explained at the end of Section. We utilized the confidence interval at the 95% level.

First, experiment-I includes performing single-step ahead forecasting. In this experiment, the proposed models read a set of n successive real measured friction coefficients and then predict a single friction coefficient. The value of n is tunable and can be adjusted to set the saving of the power consumption. For instance, if $n = 4$, then for each successive friction coefficient, four coefficients are read from the sensor, and one coefficient is predicted. This makes the power saving of the friction coefficient sensor 20%.

In experiment-II, the predictive models are trained on the real measured friction coefficients of the first m minutes of the overall running time of the UMT. Then, each forecasting model begins to forecast the friction coefficients for the remaining time of the experiment time. Thus, if the overall running time of UMT is T, then the save in the power consumption of the friction coefficient sensor is $(T - m) \times p$, where p represents the power consumption of the aforementioned sensor per minute. Of note, the predictive model is trained on the real friction coefficient of the first m minutes, but the window of the predictive model consists of the last w forecasted values, that is, no actual readings are used in the forecasting process for this experiment. Specifically, for experiment-II, we performed three experiments of the three training time intervals, $m \in \{7, 10, 13\}$.

In experiment-III, the forecasting models are trained on the frictional coefficient of the first m_1 minutes of data. Then, the predictive models forecast the friction coefficients of the next m_2 minutes while the friction coefficient sensor is not working. Then, the friction coefficient sensor is started again

Table 6: The composition of the nanofluids.

Sample	Distilled water	Palm oil (wt.%)	Graphene nanoplatelets (GNPs)
LB2000	—	—	—
PG-0.0	—	100	0 wt.%
PG-0.03	—	99.97	0.03 wt.%
PG-0.1	—	99.90	0.10 wt.%
PG-0.2	—	99.80	0.20 wt.%
PG-0.3	—	99.70	0.30 wt.%
PG-0.4	—	99.60	0.40 wt.%
W15	50 ml	—	7.5 mg concentration (0.15 mg/ml)

to produce the friction coefficients of the next m_3 minutes. Meanwhile, the predictive models are updated with the values obtained during the m_3 minutes producing no predicted values. This scenario is repeated until the end of the experiment. The observed real readings of the m_3 minutes enable the proposed model from capturing the new patterns. The goal of this experiment is to reduce the power consumption sensor and to make the accuracy of this sensor more accurate for the upcoming interval, as the sensor is turned off while the proposed model forecasts the readings. In addition, we suggested an updating mechanism for the weights of the GRU model.

Finally, in experiment-IV, as depicted in Figure 5, the proposed predictive models forecast the friction coefficient values of s seconds in advance (i.e., the brown color intervals in Figure 5). The forecasted values are obtained from the model which is trained on the real friction coefficient values (i.e., the green color intervals in Figure 5). The model utilized in this experiment can be used to provide an alert when the predicted friction coefficient value approaches a certain critical threshold that affects the product to be cut. Then, the lubrication parameters (i.e., lubricant pressure and flow rate) can be readjusted to lessen the friction before the failure of the tool and the workpiece. To obtain the most accurate results, in this experiment, the attached force sensors to the UMT pin-on-disc tribometer were not turned off like in the three previous experiments, as the main goal of this experiment is to avoid workpiece deterioration not reducing power consumption. Thus, the model is trained on real data instead of forecasted data.

To study the effect of various training set sizes on models' performance, we proposed four training and test set pairs of the original dataset. The four datasets are varied by changing the number of minutes used to collect the training data, as listed in Table 7.

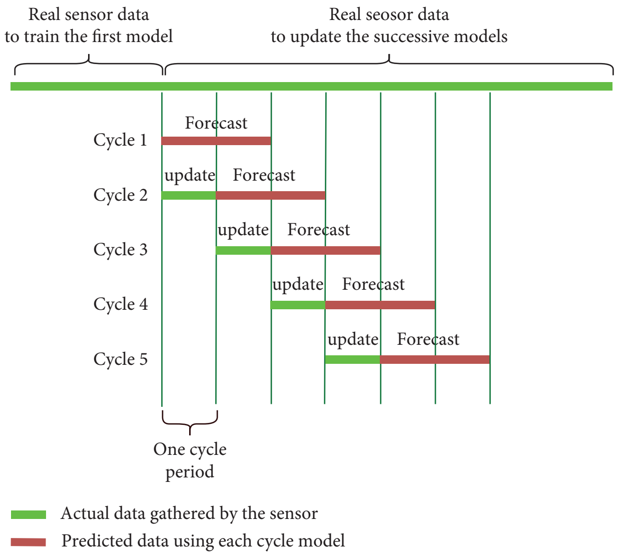

Figure 5: Experiment-IV block diagram.

5.4. Accuracy Metrics. To better assess the forecasting performance of the proposed models, we use numerous widely used time series forecasting evaluation metrics [59–61]. Therefore, four metrics are utilized, namely, mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and prediction of change in direction (POCID).

MAE measures the mean of the absolute differences between the actual values and the forecasted values, which is given by equation (12). MSE is the square value of the error between the actual values and the predicted values, which is given by equation (13). RMSE measures the root of the average squared error deviation of the predicted values with respect to the actual values, which is measured by equation (14). The lower the values for MAE, MSE, and RMSE, the better a forecasting model is.

Finally, POCID denotes the percentage of the model correctly forecasting trend (i.e., up/down) relative to the actual data trend. POCID metric is given by equation (15). A higher POCID value indicates better trend forecasting produced by the model.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \widehat{Y}_i\right|, \tag{12}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)^2, \tag{13}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)^2}, \tag{14}$$

$$\mathrm{POCID} = \frac{1}{N}\sum_{i=1}^{N} D_i \times 100\%, \tag{15}$$

Table 7: Different dataset train test split length.

Dataset	Training time interval (minutes)	Training length	Test length
Dataset I	2	12,000	76,587
Dataset II	3	18,000	70,587
Dataset III	4	24,000	64,587
Dataset IV	5	30,000	58,587

where

$$D_i = \begin{cases} 1, & \text{if } \left(Y_i - Y_{i-1}\right)\left(\widehat{Y}_i - \widehat{Y}_{i-1}\right) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

where N denotes the number of test set data records, Y_i represents the actual observation values, and $\widehat{Y}_i$ is the model output values.

5.5. Results. To mitigate the stochastic nature of the neural network, due to the initial random weight and bias values, we ran each experiment five times, whereas the average run's outcomes were reported. Moreover, we reported the mean and standard deviation of the five different runs, as shown in Figure 6. In contrast, the ARIMA model produces the same result for different runs. As a result, its error bar is a single point; thus, we ignored reporting the standard deviation of the ARIMA models.

5.5.1. Experiment-I. Running the ARIMA model for experiment-I requires training the model on the training set (e.g., observations for 2 minutes, dataset I) and forecasting a time point ahead. Then, the ARIMA model should be trained again using the same previous data points plus a new actual observation to forecast a new time point ahead, and so forth, that is, because each time the ARIMA model is trained (at each time point) without having any experience from the last trained model of the previously trained models. Additionally, fitting the ARIMA model on 2 minutes' observation, the smallest portion of the dataset to be used as training data in experiment-I takes around 52 seconds. Therefore, due to the large size of the test set, it is not applicable to train 100 ARIMA models per second to forecast the next 100 time points. By this at hand, the ARIMA model is not applicable to be utilized in experiment-I configurations.

In Figure 7, one can notice that as the k value increases, the MAE value decreases till it reaches a plateau at $k = 9$, while $k = 10$ results in the least MAE value. Therefore, we set $k = 10$ for the GRU-DNN model experiment-I and other remaining experiments. Table 8 reports the GRU-DNN model performance of experiment-I in terms of the accuracy metrics. Of note, the training and validation loss in Figure 8 is much higher than the test loss, reported in Table 8, as they are measured during the training (while data are normalized to a wider scale than the original data scale (equation (11))).

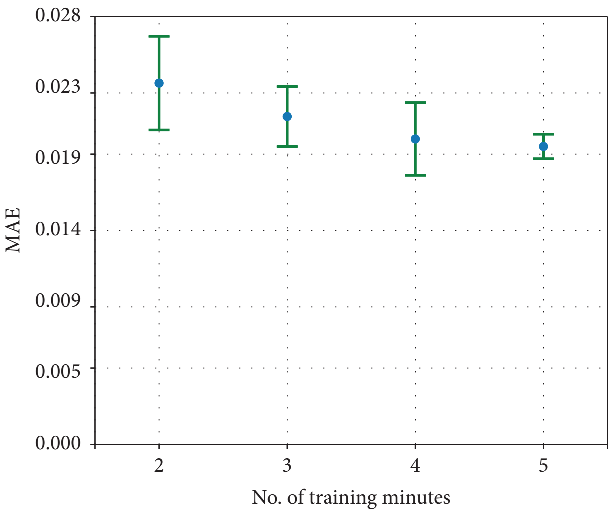

Figure 6: Error bar of 5 runs of experiment-I using dataset III for GRU-DNN model.

Figure 9 presents the first 100 values for the actual and predicted values produced by a GRU-DNN model trained on dataset III of the PG-0.1 sample. Figure 9 represents the same behavior of the other samples of Table 6. Furthermore, Figure 8 presents the loss values of the same model, i.e., training and validation loss (i.e., MAE), while other models have similar behavior. The training and validation losses confirm the model's ability to generalize to unseen data, that is, the model does not suffer from an overfitting problem.

5.5.2. Experiment-II. Table 9 presents the ARIMA results for experiment-II. In this experiment, we performed a grid search to find the best ARIMA model parameter values before fitting the model. The hyperparameter grid search shows a non-seasonality trend in the data series; therefore, $P = 0$, $D = 0$, $Q = 0$, and $S = 0$. Table 10 reports the ARIMA hyperparameters values, training time, and model sizes for experiment-II. Notably, as the training data increases, the training time, and model size increase as well, as listed in Table 11.

5.5.3. Experiment-III. The ARIMA results for experiment-III are listed in Table 12. One of the main drawbacks of the ARIMA model is that it does not provide a useful experience of fitting ARIMA on previous training.

5.5.4. Experiment-IV. Table 13 lists the values of the four used metrics for predicting 60 seconds in advance (i.e., one minute). The ARIMA model shows a better performance for all the evaluation metrics except POCID. This shows that for the scenario of predicting a set of n seconds in advance, the ARIMA model should be preferred over the GRU model. The required time to prepare the new model for forecasting the next minute is 30 and 20 seconds for the ARIMA and GRU models, respectively.

6. Discussion

The proposed GRU model is considered more suitable than the ARIMA model for the scenario of forecasting a few future friction coefficients, i.e., experiment-I, due to its time

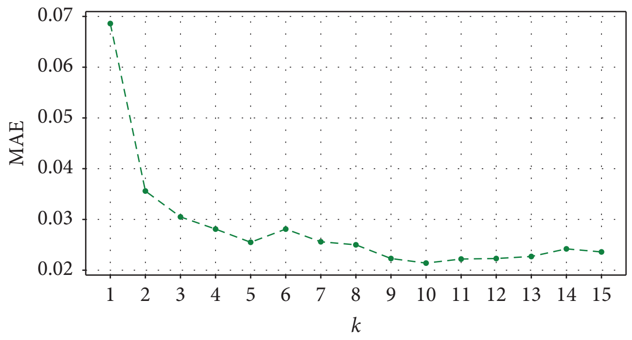

Figure 7: The MEA values of the proposed GRU-DNN model for different values of k.

Table 8: Evaluation metric scores of the proposed GRU-DNN model for experiment-I.

Dataset	MAE	MSE	RMSE	POCID
	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-2}$	%
Dataset I	4.09 ± 2.23	14.15 ± 32.71	7.79 ± 8.72	83.40 ± 9.68
Dataset II	3.57 ± 1.85	7.73 ± 16.12	6.25 ± 5.98	83.52 ± 10.49
Dataset III	3.59 ± 1.84	7.23 ± 13.97	6.21 ± 5.61	83.09 ± 10.49
Dataset IV	3.47 ± 1.65	7.20 ± 13.83	6.22 ± 5.57	84.04 ± 9.83

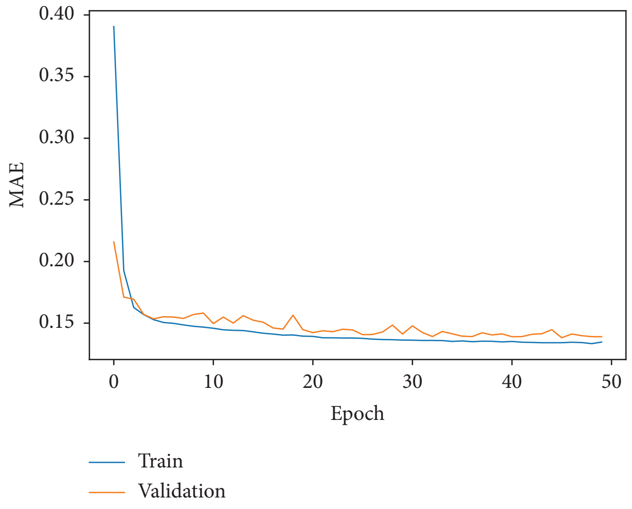

Figure 8: The proposed GRU-DNN model's training and validation loss of experiment-I using dataset III.

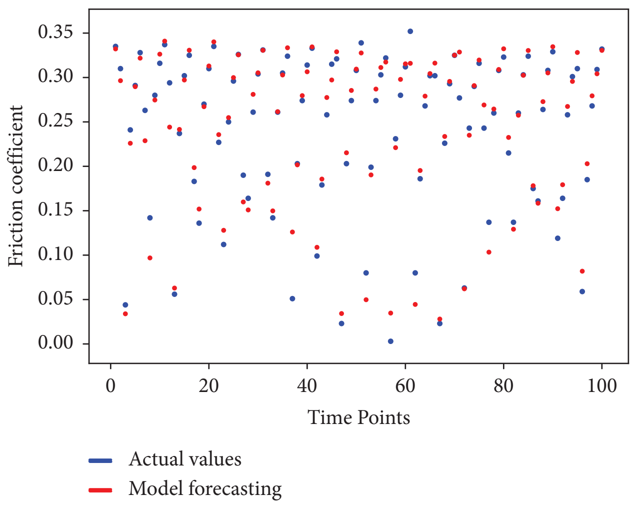

Figure 9: The actual and forecasted values of the proposed GRU-DNN model for experiment-I using dataset III of sample PG-0.1.

requirement for building the model. This scenario required updating the forecast model several times, e.g., every four friction coefficient measurements. On the other hand, the GRU model weights can be updated in fractions of a second to forecast a few friction coefficients. For the task of forecasting an extended number of friction coefficients, and measurements for tens of seconds, the ARIMA model outperforms the GRU model in all evaluation metrics but not in the POCID. Thus, the ARIMA model can predict the friction coefficient for a prolonged period of time in advance better than the proposed GRU model while the GRU model predicts the changes in the friction coefficient over a short period of time better, e.g., experiment-I.

Regarding the predictive model requirements, the memory and time requirements of the GRU-DNN model for experiments I and II are listed in Tables 11 and 14, respectively. The listed memory sizes, training, and forecast times show that the proposed GRU-DNN model requires fewer resources relative to the ARIMA model. For instance, comparing the ARIMA and GRU models' memory for experiment-II, the ARIMA model's size is four orders of magnitude larger while the forecasting time of the ARIMA is smaller than the GRU-DNN model by an order of magnitude. For training and forecasting times, the proposed ARIMA model outperformed the GRU-DNN model in terms, as listed in Table 9. Based on this discussion, in experiment-II, the ARIMA and proposed GRU-DNN models achieved very close results where the ARIMA model slightly performed better. Thus, the ARIMA model is preferred over the proposed GRU-DNN for scenarios similar to experiment-II.

In experiment-III and experiment-IV, the performance of the ARIMA model is clearly better, as the error metric values are lower for the ARIMA model. The listed results of Table 13 are for forecasting one minute in advance. Thus, comparing Tables 12 and 13 for one minute forecasting only outlines a better performance for experiment-IV. This can be linked to the fact that the predictive models used in experiment-IV are trained on real sensors' readings, as the force sensors are not turned off during experiment-IV. For experiment-III, the predictive models are trained on mixed data of real sensors' readings and forecasted readings; this is because the force sensors were turned off during the forecasting period to reduce the power consumption. As the forecasted readings include some errors, these errors affect the predictive model performance.

Table 9: Evaluation metric scores of the proposed models for experiment-II.

Model	Training minutes	MAE	MSE	RMSE	POCID
		$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-2}$	%
ARIMA	7	**9.22 ± 3.58**	**18.52 ± 21.43**	**11.96 ± 6.33**	**32.22 ± 21.07**
	10	**8.89 ± 3.18**	**15.94 ± 14.44**	**11.45 ± 5.18**	**31.38 ± 22.68**
	13	**8.84 ± 3.14**	**17.37 ± 17.98**	**11.76 ± 5.80**	**39.80 ± 18.96**
GRU-DNN	7	9.30 ± 3.00	19.87 ± 16.37	12.91 ± 5.52	47.13 ± 16.48
	10	9.36 ± 3.43	18.46 ± 12.77	12.57 ± 5.02	50.38 ± 7.79
	13	9.55 ± 3.19	20.11 ± 15.98	13.02 ± 5.46	48.12 ± 16.75

Bold values indicate best results achieved for each experiment.

Table 10: ARIMA parameters for experiment-II.

Experiment training minutes	Hyperparameters			Training time (seconds)	Forecasting time (seconds)	Model size (MB)
	p	d	q			
7	5	0	1	84	1.0	191
10	5	0	1	109	0.7	273
13	5	0	4	183	0.3	355

Table 11: The proposed GRU-DNN model performance for experiment-II.

Training minutes	No. of epoch	Training time (sec)	Forecasting time (sec)	Model size (MB)
7	28	122.3	55.8	0.031
10	22	219.2	35.7	0.031
13	28	429.0	13.4	0.033

Table 12: Evaluation metric scores of the proposed models for experiment-III.

Mode	Training minutes	Forecast	MAE	MSE	RMSE	POCID
			$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-2}$	%
ARIMA	0.5	0.25	**7.54 ± 2.93**	**29.39 ± 61.90**	**9.99 ± 5.44**	**53.19 ± 13.79**
	0.5	1.0	**8.88 ± 4.25**	**34.66 ± 73.22**	**11.58 ± 7.55**	**45.75 ± 13.13**
	1.0	0.5	**7.83 ± 2.98**	**27.81 ± 55.73**	**10.44 ± 5.79**	51.69 ± 11.12
	1.0	2.0	**8.94 ± 4.21**	**31.45 ± 63.10**	**11.73 ± 7.64**	**48.06 ± 9.93**
GRU-DNN	0.5	0.25	10.07 ± 5.53	38.65 ± 65.11	12.88 ± 7.14	45.58 ± 24.33
	0.5	1.0	10.02 ± 5.04	31.84 ± 42.59	13.67 ± 7.18	44.99 ± 14.38
	1.0	0.5	8.33 ± 3.47	29.73 ± 51.60	11.44 ± 5.69	53.80 ± 16.73
	1.0	2.0	9.40 ± 3.36	31.53 ± 53.54	13.46 ± 7.18	46.80 ± 13.38

Bold values indicate best results achieved for each experiment.

Table 13: Evaluation metric scores of the proposed models for experiment-IV.

Model	MAE	MSE	RMSE	POCID
	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-2}$	%
ARIMA	**7.75 ± 2.77**	**21.23 ± 36.38**	**10.29 ± 5.28**	40.01 ± 11.00
GRU-DNN	9.82 ± 4.18	84.40 ± 204.88	13.64 ± 8.61	**47.86 ± 9.98**

Bold values indicate best results for achieved each experiment.

Table 14: The proposed GRU-DNN model performance for experiment-I.

Dataset	No. of epoch	Training time (sec)	Forecasting time (sec)	Model size (MB)
Dataset I	26	54.4	3.5	0.040
Dataset II	40	216.9	3.0	0.031
Dataset III	37	160.5	2.9	0.042
Dataset IV	42	164.7	2.2	0.040

6.1. Applications of the Proposed Predictive Model. There are two possible applications of the proposed work. The main target of the study is to predict the friction based on real-time readings of the friction coefficients and control the lubrication parameters like the fluid mist pressure to keep the friction coefficients at minimum levels. Achieving this goal leads to a significant reduction in energy consumption. The proposed technique can be applied directly to the field of manufacturing whereas cutting processes are considered a heavy energy consumption sector and friction is the main suspect. The second possible application is to utilize the proposed model in reducing or avoiding surface thermal damage. By predicting the high level of friction coefficients in advance, it is possible to readjust the lubrication parameters (fluid pressure, fluid flow rate, etc.) to maintain lower ranges of friction coefficients, as high ranges of friction coefficients may deteriorate the workpiece.

7. Conclusion

The task of monitoring the friction coefficient of the surface friction is vital in any metal cutting technique, but it is a time power consuming as well. Besides, the ability to predict future friction coefficients from historical data is a vital task to reduce or avoid thermal damage. In this work, we proposed the first predictive models to capture the patterns of the friction coefficient during a metal cutting process. We generated a real friction coefficient of the surface friction using UMT. Then, we proposed ARIMA and GRU-DNN models to perform the forecast task on a real dataset. The ARIMA and GRU-DNN model parameters are tuned to get the best performance. Finally, the proposed models are tested on four different power consumption reduction scenarios, and the fourth scenario proves the ability of the proposed model to avoid thermal damage. The proposed models show a significant performance in terms of prediction accuracy. The future directions include using a hybrid model of the ARIMA and GRU-DNN models using ensemble learning models to combine the results of these two models. Besides, increasing the dataset by collecting friction coefficient data from different milling machines might increase the prediction accuracy rates.

Data Availability

The utilized datasets and the source code of this work are freely accessible online on the author's GitHub website: https://github.com/Ahmed-Fathalla/Friction_coef-forecasting.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Ahmad Salah and Ahmed Fathalla contributed equally to this work.

Acknowledgments

This study was supported via funding from Prince Sattam Bin Abdulaziz University (project number PSAU/2023/R/1445).

References

[1] A. M. M. Ibrahim, W. Li, H. Xiao, Z. Zeng, Y. Ren, and M. S. Alsoufi, "Energy conservation and environmental sustainability during grinding operation of Ti–6Al–4V alloys via eco-friendly oil/graphene nano additive and Minimum quantity lubrication," *Tribology International*, vol. 150, Article ID 106387, 2020.

[2] I. Lazoglu and Y. Altintas, "Prediction of tool and chip temperature in continuous and interrupted machining," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 9, pp. 1011–1022, 2002.

[3] F. Klocke, W. König, and K. Gerschwiler, "Advanced machining of titanium-and nickel-based alloys," in *Advanced Manufacturing Systems and Technology*, pp. 7–21, Springer, Berlin, Germany, 1996.

[4] S. Ghosh, S. Ghosh, and P. Venkateswara Rao, "Application of sustainable techniques in metal cutting for enhanced machinability: a review," *Journal of Cleaner Production*, vol. 100, pp. 17–34, 2015.

[5] A. M. M. Ibrahim, X. Shi, W. Zhai, and K. Yang, "Improving the tribological properties of nial matrix composites via hybrid lubricants of silver and graphene nano platelets," *RSC Advances*, vol. 5, no. 76, pp. 61554–61561, 2015.

[6] N. A. Bassiouny, M. Al-Makky, and H. Youssef, "Parameters affecting the quality of friction drilled holes and formed thread in austenitic stainless steel aisi 304," *The International Journal of Advanced Manufacturing Technology*, vol. 125, no. 3-4, pp. 1493–1509, 2023.

[7] D. Ambrosio, A. Tongne, V. Wagner, G. Dessein, and O. Cahuc, "Towards material flow prediction in friction stir welding accounting for mechanisms governing chip formation in orthogonal cutting," *Journal of Manufacturing Processes*, vol. 85, pp. 450–465, 2023.

[8] J. Zhang, Z. Liu, and J. Du, "Prediction of cutting temperature distributions on rake face of coated cutting tools," *The International Journal of Advanced Manufacturing Technology*, vol. 91, no. 1-4, pp. 49–57, 2017.

[9] B. Zhao, Z.-N. Zhang, and X.-D. Dai, "Prediction of wear at revolute clearance joints in flexible mechanical systems," *Procedia Engineering*, vol. 68, pp. 661–667, 2013.

[10] H. Fathipour-Azar, "Mean cutting force prediction of conical picks using ensemble learning paradigm," *Rock Mechanics and Rock Engineering*, vol. 56, no. 1, pp. 221–236, 2023.

[11] M. Akram and C. El, "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks," *International Journal of Computers and Applications*, vol. 143, no. 11, pp. 7–11, 2016.

[12] Z. Alameer, A. Fathalla, K. Li, H. Ye, and Z. Jianhua, "Multistep-ahead forecasting of coal prices using a hybrid deep learning model," *Resources Policy*, vol. 65, Article ID 101588, 2020.

[13] A. Fathalla, A. Salah, K. Li, K. Li, and P. Francesco, "Deep end-to-end learning for price prediction of second-hand items," *Knowledge and Information Systems*, vol. 62, no. 12, pp. 4541–4568, 2020.

[14] A. Salah, M. Bekhit, E. Eldesouky, A. Ali, and A. Fathalla, "Price prediction of seasonal items using time series analysis," *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 445–460, 2023.

[15] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep lstm recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[16] A. Ali, A. Fathalla, A. Salah, M. Bekhit, and E. Eldesouky, “Marine data prediction: an evaluation of machine learning, deep learning, and statistical predictive models,” *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 8551167, 13 pages, 2021.

[17] Y. Song, H. Cao, D. Qu et al., “Impact effect-based dynamics force prediction model of high-speed dry milling ud-cfrp considering size effect,” *International Journal of Impact Engineering*, vol. 179, Article ID 104659, 2023.

[18] S. Saha, A. S. Kumar, G. Malayath, S. Deb, and P. P. Bandyopadhyay, “Energy balance model to predict the critical edge radius for adhesion formation with tool wear during micro-milling,” *Journal of Manufacturing Processes*, vol. 93, pp. 219–238, 2023.

[19] Z. Gu, S. Pang, Y. Li, Q. Li, and Y. Zhang, “Turbo-fan engine acceleration control schedule optimization based on dnn-lpv model,” *Aerospace Science and Technology*, vol. 128, Article ID 107797, 2022.

[20] W. Geng, D. Lv, X. Yu, and Y. Wang, “Prediction model optimization and experimental verification based on the material characteristic and dynamic angle on thrust force and torque in drilling gjv450,” *The International Journal of Advanced Manufacturing Technology*, vol. 126, no. 11-12, pp. 5571–5582, 2023.

[21] U. M. R. Paturi, S. T. Palakurthy, and N. Reddy, “The role of machine learning in tribology: a systematic review,” *Archives of Computational Methods in Engineering*, vol. 30, no. 2, pp. 1345–1397, 2023.

[22] A. Saravanan, S. Parida, M. Murugan, M. S. Reddy, P. Elumalai, and S. Kumar Dash, “Thermal performance prediction of a solar air heater with a c-shape finned absorber plate using rf, lr and knn models of machine learning,” *Thermal Science and Engineering Progress*, vol. 38, Article ID 101630, 2023.

[23] A. Singh, M. Wolf, G. Jacobs, and F. König, “Machine learning based surrogate modelling for the prediction of maximum contact temperature in ehl line contacts,” *Tribology International*, vol. 179, Article ID 108166, 2023.

[24] P. Stuhler and N. Nagler, “Smearing in full complement roller bearings: parameter study and damage analysis,” *Proceedings of the Institution of Mechanical Engineers- Part J: Journal of Engineering Tribology*, vol. 236, no. 12, pp. 2535–2546, 2022.

[25] A. Mishra, “Artificial intelligence algorithms for prediction of the ultimate tensile strength of the friction stir welded magnesium alloys,” *International Journal on Interactive Design and Manufacturing*, pp. 1–9, 2023, https://link.springer.com/article/10.1007/s12008-022-01180-w#citeas.

[26] A. H. Elsheikh, “Applications of machine learning in friction stir welding: prediction of joint properties, real-time control and tool failure diagnosis,” *Engineering Applications of Artificial Intelligence*, vol. 121, Article ID 105961, 2023.

[27] Y. Yang, B. Hao, X. Hao et al., “A novel tool (single-flute) condition monitoring method for end milling process based on intelligent processing of milling force data by machine learning algorithms,” *International Journal of Precision Engineering and Manufacturing*, vol. 21, no. 11, pp. 2159–2171, 2020.

[28] J. Diaz-Rozo, C. Bielza, and P. Larrañaga, “Machine learning-based cps for clustering high throughput machining cycle conditions,” *Procedia Manufacturing*, vol. 10, pp. 997–1008, 2017.

[29] P. Krishnakumar, K. Rameshkumar, and K. Ramachandran, “Acoustic emission-based tool condition classification in a precision high-speed machining of titanium alloy: a machine learning approach,” *International Journal of Computational Intelligence and Applications*, vol. 17, no. 03, Article ID 1850017, 2018.

[30] G. Lawrance, P. S. Paul, J. Mohammed, M. Gunasegeran, and P. E. Sudhagar, “Prediction of cutting performance using artificial neural network during buffered impact damper-assisted boring process,” *Multiscale and Multidisciplinary Modeling, Experiments and Design*, vol. 6, no. 4, pp. 671–684, 2023.

[31] R. Oberlé, S. Schorr, L. Yi, M. Glatt, D. Bähre, and J. C. Aurich, “A use case to implement machine learning for life time prediction of manufacturing tools,” *Procedia Corporate Insolvency Resolution Process*, vol. 93, pp. 1484–1489, 2020.

[32] A. Gouarir, G. Martínez-Arellano, G. Terrazas, P. Benardos, and S. Ratchev, “In-process tool wear prediction system based on machine learning techniques and force analysis,” *Procedia CIRP*, vol. 77, pp. 501–504, 2018.

[33] S. Tanvir and L. Qiao, “Surface tension of nanofluid-type fuels containing suspended nanomaterials,” *Nanoscale Research Letters*, vol. 7, pp. 226–310, 2012.

[34] S. Vafaei, A. Purkayastha, A. Jain, G. Ramanath, and T. Borca-Tasciuc, “The effect of nanoparticles on the liquid–gas surface tension of bi2te3 nanofluids,” *Nanotechnology*, vol. 20, no. 18, Article ID 185702, 2009.

[35] M. Bhuiyan, R. Saidur, R. Mostafizur, I. Mahbubul, and M. Amalina, “Experimental investigation on surface tension of metal oxide–water nanofluids,” *International Communications in Heat and Mass Transfer*, vol. 65, pp. 82–88, 2015.

[36] G. Astm, *133 Standard Test Method for Linearly Reciprocating ball-on-flat Sliding Wear*, ASTM International, West Conshohocken, PA, USA, 2016.

[37] S. Debnath, M. M. Reddy, and Q. S. Yi, “Environmental friendly cutting fluids and cooling techniques in machining: a review,” *Journal of Cleaner Production*, vol. 83, pp. 33–47, 2014.

[38] G. E. Box and G. M. Jenkins, “Some recent advances in forecasting and control,” *Applied Statistics*, vol. 17, no. 2, pp. 91–109, 1968.

[39] G. J. Box, *Gm Time Series Analysis*, Forecasting and Control Holden-Day, San Francisco, CA, USA, 1970.

[40] L. Xiao, Y. Zhang, K. Li, B. Liao, and Z. Tan, “A novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion,” *Neurocomputing*, vol. 331, pp. 483–492, 2019.

[41] C. Chen, K. Li, S. G. Teo et al., “Gated residual recurrent graph neural networks for traffic prediction,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 485–492, 2019.

[42] Z. Quan, X. Lin, Z.-J. Wang, Y. Liu, F. Wang, and K. Li, “A system for learning atoms based on long short-term memory recurrent neural networks,” in *Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 728–733, IEEE, Madrid, Spain, December, 2018.

[43] K. Cho, B. Van Merriënboer, C. Gulcehre et al., “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” 2014, https://arxiv.org/abs/1406.1078.

[44] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light gated recurrent units for speech recognition,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[46] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[49] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a python library for model selection and hyperparameter optimization," *Computational Science and Discovery*, vol. 8, no. 1, Article ID 014008, 2015.

[50] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: a python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*, pp. 13–20, Citeseer, Austin, TX, USA, July 2013.

[51] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, USA, June 2010.

[52] S. Seabold and J. Perktold, "Statsmodels: econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, Austin, TX, USA, June 2010.

[53] M. Abadi, P. Barham, J. Chen et al., "Tensorflow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design And Implementation (OSDI 16)*, pp. 265–283, Savannah, GA, USA, November 2016.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[55] T. E. Oliphant, *A Guide to NumPy*, vol. 1, Trelgol Publishing, New york, NY, USA, 2006.

[56] J. D. Hunter, "Matplotlib: a 2d graphics environment," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[57] A. M. M. Ibrahim, X. Shi, A. Zhang, K. Yang, and W. Zhai, "Tribological characteristics of nial matrix composites with 1.5 wt% graphene at elevated temperatures: an experimental and theoretical study," *Tribology Transactions*, vol. 58, no. 6, pp. 1076–1083, 2015.

[58] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, pp. 9–48, Springer, Berlin, Germany, 2012.

[59] M. C. A. Neto, G. Tavares, V. M. Alves, G. D. Cavalcanti, and T. I. Ren, "Improving financial time series prediction using exogenous series and neural networks committees," in *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Barcelona, Spain, July, 2010.

[60] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A systematic review of fundamental and technical analysis of stock market predictions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 3007–3057, 2019.

[61] B. Yang, W. Bao, and Y. Chen, "Time series prediction based on complex-valued s-system model," *Complexity*, vol. 2020, Article ID 6393805, 13 pages, 2020.